VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
FACULTY OF MECHANICAL ENGINEERING

ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY
INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

VIRTUÁLNÍ ZPROVOZNĚNÍ ROBOTICKÉHO PRACOVIŠTĚ PRO MANIPULACI S VÝROBKEM

VIRTUAL COMMISSIONING OF THE ROBOTIC WORKPLACE FOR WORKPIECE MANIPULATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR
Bc. Tomáš Lorenc

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. Jakub Bražina

BRNO 2021

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadaní diplomové práce

Ústav:	Ústav výrobních strojů, systémů a robotiky
Student:	**Bc. Tomáš Lorenc**
Studijní program:	Strojní inženýrství
Studijní obor:	Výrobní stroje, systémy a roboty
Vedoucí práce:	**Ing. Jakub Bražina**
Akademický rok:	2020/21

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Virtuální zprovoznění robotického pracoviště pro manipulaci s výrobkem

Stručná charakteristika problematiky úkolu:

V rámci práce se student seznámí s problematikou virtuálního zprovoznění robotického pracoviště. Student navrhne chybějící komponenty a vytvoří simulační model pracoviště ve vhodném softwaru. Následuje návrh robotických a PLC programů a virtuální zprovoznění.

Cíle diplomové práce:

Rešerše dané problematiky.
Konstrukční návrh koncového efektoru robotu.
Tvorba simulačního modelu pracoviště.
Příprava robotických a PLC programů.
Virtuální zprovoznění.
Doporučení pro praxi.

Seznam doporučené literatury:

SICILIANO, Bruno a Oussama KHATIB. Springer Handbook of Robotics. Berlin: Springer, 2008. ISBN 978-3-540-23957-4.

KOLÍBAL, Zdeněk. Roboty a robotizované výrobní technologie. Brno: Vutium, 2016. ISBN 978-8--2144-828-5.

NOF, Shimon. Springer Handbook of Automation. Berlin: Springer, 2009. ISBN 978-3-540-78830-0.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2020/21

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D. ředitel ústavu	doc. Ing. Jaroslav Katolický, Ph.D. děkan fakulty

ABSTRAKT

Diplomová práce se zabývá problematikou virtuálního zprovoznění pracoviště a zprovozněním komunikačních kanálů k tomu potřebných. V teoretické části popisuje přínosy virtuálního zprovoznění, programování robotů a PLC a koncové efektory robotů. V praktické části se zabývá návrhem koncového efektoru pro robot, virtuálním zprovozněním pomocí programů Process Simulate a TwinCAT 3, komunikací mezi programy a tvorbou programů pro pracoviště.

ABSTRACT

The diploma thesis deals with the issue of virtual commissioning of the workplace and the commissioning of communication channels needed for this purpose. The theoretical part describes the benefits of virtual commissioning, robot and PLC programming and effectors for robots. The practical part deals with the design of the final robot effector, virtual commissioning using the programs Process Simulate and TwinCAT 3, communication between programs and creation of programs for the workplace.

KLÍČOVÁ SLOVA

Virtuální zprovoznění, Tecnomatix Process Simulate, TwinCAT 3, OPC UA, TCP/IP, Beckhoff PLC, KUKA robot.

KEYWORDS

Virtual commissioning, Tecnomatix Process Simulate, TwinCAT 3, OPC UA, TCP/IP, Beckhoff PLC, KUKA robot.

BIBLIOGRAFICKÁ CITACE

LORENC, Tomáš. *Virtuální zprovoznění robotického pracoviště pro manipulaci s výrobkem*. Brno, 2021. Dostupné také z: https://www.vutbr.cz/studenti/zav-prace/detail/132992. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav výrobních strojů, systémů a robotiky. Vedoucí práce Jakub Bražina.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu práce Ing. Jakubu Bražinovi za vedení při psaní práce a za ochotu a čas věnovaný konzultacím. Dále bych rád poděkoval lidem z ústavu ÚVSSR za pomoc a cenné rady při realizaci dílčích částí práce.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Jakuba Bražiny a s použitím literatury uvedené v seznamu.

V Brně dne 16.5.2021 ……

Lorenc Tomáš

OBSAH

1 ÚVOD

Výroba se v dnešní době stává stále častěji doménou automatizovaných výrobních linek a robotických pracovišť. Je již jen velice málo odvětví, ve kterých při vzniku výrobku není použit žádný automatizační prvek. Není tedy neobvyklé se setkat s robotem i v menší firmě, což ještě před několika lety bylo spíše vzácností. U velkých výrobních závodů, zaměřených především na sériovou či dokonce hromadnou výrobu je pak naprostou nutností snažit se veškerou možnou opakující se práci svěřit robotům. Kromě nich se navíc v těchto výrobách uplatňují celé výrobní linky, obsahující řadu dalších automatizačních prvků, jak jsou servopohony, pneumatické prvky, snímače a v neposlední řadě se v poslední době rozšiřuje také použití kamer. Všechny tyto prvky pak vyžadují nějaké řízení, jelikož samy o sobě jsou pouze kusy železa a plastu. O toto řízení se starají řídící systémy robotů, jednotlivých složitějších komponent a PLC.

Aby však bylo možné všechny tyto prvky takzvaně oživit, je potřeba pro tyto řídící systémy vytvořit pokyny v podobě programu. Jelikož je však velká spousta různých výrobců hardwaru i řídících systémů, tvorba programů není jednoduchá, jelikož každý dodavatel komponent má vlastní software pro jejich nastavení a kalibraci, každý výrobce PLC má vlastní vývojové prostředí, které navíc bývá většinou zpoplatněno a každý výrobce průmyslových robotů má kromě svého vývojového prostředí navíc vlastní programovací jazyk. Často tedy bývá značně složité naprogramovat nejenom celou výrobní linku, ale i menší pracoviště, které využívá různou škálu prvků od různých výrobců. Navíc se vše většinou zkouší až na postavené výrobní lince nebo pracovišti, takže na případné chyby se přichází až po nahrání softwaru do řídícího systému a spuštění programu. To bývá časově a tedy i finančně náročné. Proto se v poslední době začíná především u větších firem používat virtuální zprovoznění pracovišť. Tento způsob programování šetří nejenom čas a peníze, ale především umožňuje zjištění a vyhodnocení chyb již při vývoji softwaru, ne až při jeho testování na skutečném stroji.

Tato diplomová práce se zabývá právě problematikou virtuálního zprovoznění, avšak přibližuje také konvenční metody programování a problematiku řídících systémů, PLC a robotů. Virtuální zprovoznění je provedeno na výukovém pracovišti Ústavu výrobních strojů, systémů a robotiky fakulty strojního inženýrství Vysokého učení technického v Brně.

2 MOTIVACE

Motivací práce je kromě vytvoření simulačního modelu a virtuálního zprovoznění pracoviště také zjištění hranic simulace pro určité typy komunikace a propojení s reálným pracovištěm. Konkrétně se jedná o propojení a použití informací, které řídící PLC dostává z kamerového systému pracoviště, při simulaci pracoviště v programu Process Simulate.

V dostupných zdrojích tento způsob řízení simulace nebyl nalezen, proto je cílem zjištění, zda lze za použití současných verzí softwarů toto řízení provést a v případě úspěchu popsat postup řešení.

V případě, že by toto řízení možné nebylo, je cílem práce návrh řešení simulace do podoby, která pomůže zrychlit návrh a programování reálného pracoviště, popis překážek řízení simulace za pomocí údajů z kamery a případný návrh řešení do budoucna.

3 PŘEHLED SOUČASNÉHO STAVU POZNÁNÍ

Teoretické znalosti jsou nezbytné pro správné pochopení problematiky a následné konstrukční, softwarové a programovací návrhy. V případě této práce se jedná především o problematiky robotů, jejich řízení a koncových efektorů. Dále pak řízení pracovišť pomocí PLC a v neposlední řadě také chápání důležitosti virtuálního zprovoznění.

3.1 Robotické pracoviště

S výjimkou jednoduchých montážních pracovišť, je robotické pracoviště nejčastějším druhem automatizace, se kterým se dnes můžeme setkat. Typů pracovišť je nepřeberné množství, vždy však využívají jednoho nebo více průmyslových robotů. Pracoviště může mít svůj konkrétní účel bez návaznosti na ostatní části výrobní linky. V takovém případě se většinou pracoviště sestává pouze z robota, nářadí potřebného pro danou činnost, zařízení pro uchopení zpracovávaného předmětu a především bezpečného vstupu předmětu do operačního prostoru robotu. Výrobek pak robotu většinou předává člověk. Proces předání musí být naprosto bezpečný a člověk se nesmí dostat do kontaktu s robotem. Obvykle se k tomuto účelu používá různých otočných stolů s přepážkou, která odděluje operátora od robotu.

Obzvláště při zakomponování pracoviště do větší výrobní linky však bývá práce člověka úplně eliminována. Výrobek je na pracoviště dopraven pomocí nějaké dopravního systému, ať už se jedná o jednoduchý dopravník nebo speciálně vyvinutou platformu, jak tomu bývá v automobilovém průmyslu. V automotive mají také robotická pracoviště největší zastoupení. Tržní síla automobilek jim umožňuje mohutně investovat do technologií a automatizace. Díky tomu je tento druh průmyslu největším tahounem automatizace a nejvíce nutí automatizační firmy přicházet s novými řešeními, druhy robotických pracovišť a spoluprací různých druhů technologií.

Zvýše uvedeného vyplývá, že robotická pracoviště jsou doménou především výrobní oblasti, případně následného distribučního řetězce. Je to dáno především nutností neměnných okolních podmínek, jelikož programy robotů i nadřazených PLC bývají striktně dané a provádějí pouze před programovanou rutinu. To se sice postupně začíná měnit se stále větším zapojení kamer a zlepšujícími se algoritmy pro rozpoznání obrazu, založenými na neuronových sítích, nicméně jsme zatím stále daleko od autonomního výrobního provozu, schopného reagovat na neočekávané změny, pro které nemá řídící systém připravenou reakci.

Jisté však je, že použití robotů se bude v budoucnu stupňovat a budu lidskou práci postupně nahrazovat i při činnostech, které se ještě nedávno zdály nepravděpodobné, jako například bin picking operace na obr. 1 [1].

OBR. 1 BIN PICKING OPERACE

3.2 Druhy robotických pracovišť

3.2.1 Manipulační pracoviště

Jedná se o nejčastější pracoviště využívající průmyslové roboty. Manipulace s obrobky (operace se někdy také nazývá pick and place) totiž patří obecně k jednodušším úkonům, které mohou roboty vykonávat.

Z konstrukčního hlediska je nutné navrhnout přípravky pro uložení dílu, vybrat ideální způsob koncového efektoru a správně ho nadimenzovat. Dalším kritickým bodem je výběr robotu s ohledem na velikost dílu, požadovanou rychlost a přesnost. V potravinářském a farmaceutickém průmyslu jsou proto často používány roboty typu scara nebo roboty s paralelní kinematickou strukturou, jelikož předměty pro manipulaci mají malou hmotnost a důraz je kladen spíše na rychlou manipulaci. Oproti tomu v těžkém průmyslu, například v ocelárnách, jsou upřednostňovány roboty s lineární kinematickou strukturou a velkou nosností. Typický příklad robotické manipulace s výrobkem je na obr. 2 [2].

Z programátorského hlediska je manipulace také jednodušším úkonem, jelikož tzv. pick and place operace většinou neobsahuje složité křivky pohybu a vystačí si lineárními nebo point to point pohyby z bodu do bodu. Další příkazy se pak týkají uchopení a puštění manipulovaného dílu, případně kontroly přítomnosti dílu pomocí senzoru. Složitějšími se manipulace stává až v případě spolupráce robota s kamerou, kdy například po dopravníkovém pásu jede řada neorientovaných předmětů, kamera musí detekovat jejich přítomnost, orientaci a informace poslat řídícímu systém. Obzvlášť obtížná je pak takováto pick and place operace, pokud robot sbírá předměty z jedoucího dopravníkového pásu, kdy musí být brána v potaz také rychlost pásu, která se nejčastěji zjišťuje pomocí enkodéru. Tento způsob manipulace se často používá právě v potravinářském a farmaceutickém průmyslu. Pro dostačující produktivitu je nutné, aby robot pracoval dost rychle, proto se s pick and place operací s kamerovým viděním moc nesetkáváme při výrobě větších a těžších dílů, kdy je výhodnější výrobek správě orientovat a detekovat jiným způsobem a robotem použít robot s dostatečnou nosností.

OBR. 2 ROBOT PŘI MANIPULACI

3.2.2 Další typy pracovišť

Svařovací pracoviště

Dalším často používaným typem pracoviště je svařovací pracoviště. Svařovací operace patří na opačný konec stupnice obtížnosti než obyčejná manipulace s dílem. Prvním náročnějším bodem je koncový efektor. Není jej možné jednoduše navrhnout a přizpůsobit každému výrobku individuálně, jelikož svařovací koncový efektor musí být kompatibilní i se zbylým svařovacím vybavením. Mezi toto vybavení patří zdroj (v ideálním případě jako propojený systém s robotem a řídícím systémem), podavač drátu a hořák. [3] Výrobci robotů většinou spolupracují s výrobci svářeček a nejvýhodnější způsob návrhu svařovacího pracoviště je po konzultaci s dodavatelem robotu použít jím doporučené řešení.

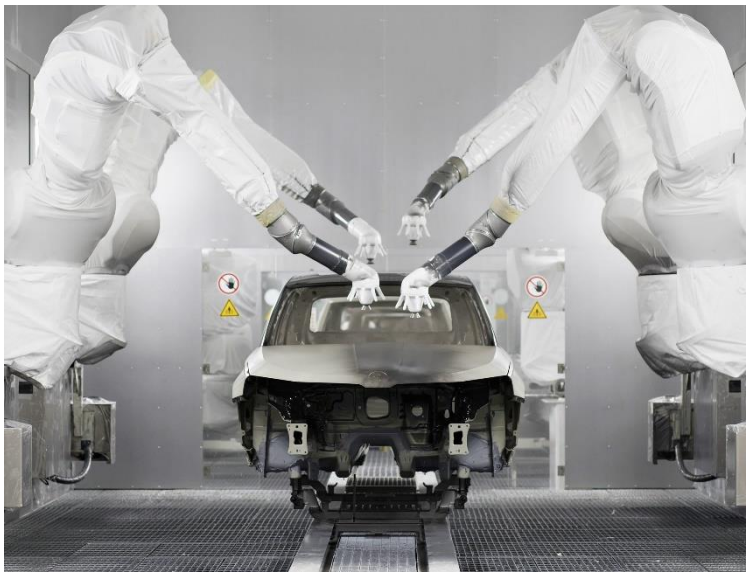
Z programátorského hlediska pak svařovací operace patří k náročnějším, jelikož je důležitá přesnost navedení robotu do místa svaru a svary často mají složité kontury, po kterých musí robot jet. Navíc se k pohybu přidává i řízení samotného svařování. Je potřeba kontrolovat začátek a konec svařování, správné použití svařovacích programů a v neposlední řadě bývají také automatizovány všechny přídavné operace svařovacího efektoru, jako například čištění.

Lakovací pracoviště

Lakovací pracoviště většinou bývají k vidění ve větších podnicích, jako jsou automobilky a používají se pro větší objemy výroby. Jelikož nalakování složitějších dílů, jako je například karoserie automobilu, většinou nemůže obsáhnout pouze jeden robot, probíhá lakování většinou v režimu lakovací linky. Z pohledu programování pak je potřeba zkoordinovat pohyby jednotlivých robotů a zajistit, aby nedocházelo ke kolizím. Příklad lakovacího pracoviště je na obr. 3 [4].

Ostatní pracoviště

Robotických operací a od nich odvozených pracovišť je samozřejmě celá řada. Mezi další běžně používané typy patří pracoviště pro aplikaci různých látek, jako je například nanášení lepidla, dále pak obráběcí nebo měřící pracoviště.



OBR. 3 LAKOVACÍ PRACOVIŠTĚ

3.3 Roboty se sériovou kinematikou

Typů robotů existuje celá řada. Mezi známější patří roboty s paralelní kinematickou strukturou, roboty typu Scara nebo roboty s translačními vazbami. Nejznámějším a také nejpoužívanějším typem robotů jsou však roboty se sériovou kinematickou strukturou. V průmyslu je nejčastěji používán konkrétně typ s rotačními kinematickými dvojicemi. Používány jsou především pro svoji univerzalitu a relativně jednoduché řízení v porovnání například s roboty s paralelní kinematickou strukturou.

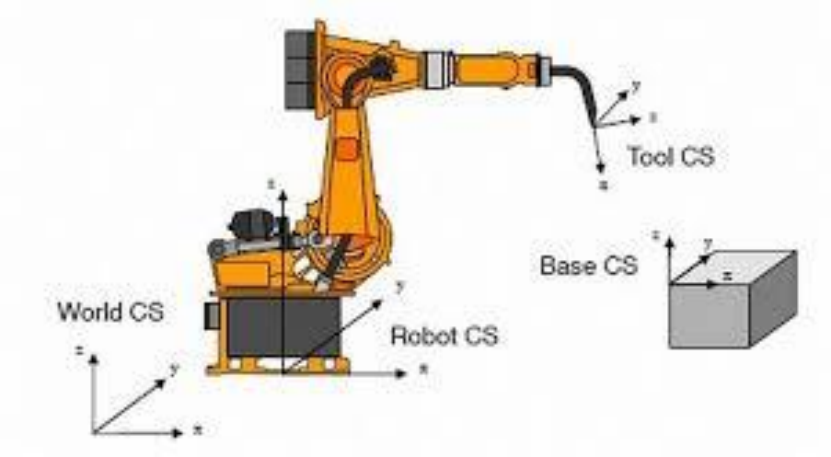
Z názvu kinematiky vyplývá, že rotační vazby jsou skládány za sebe, při čemž nejčastěji bývá použito šest rotačních vazeb, vzniká tedy šest os rotace. Díky tomu se průmyslovým robotům také říká šestiosé roboty. Pohyb koncového bodu robotu je výsledkem součtu pohybů všech předchozích pohybů.

3.3.1 Souřadné systémy

Souřadný systém Svět (World)
Jedná se o hlavní kartézský souřadný systém robotu. Souvisí s ním kartézský souřadný systém, který je umístěn v patě robotu (u robotů ABB je nazýván jako Robot Base Coordinate System a u robotů KUKA ROBROOT). Společně tyto dva systémy určují polohu a orientaci robotu vůči okolnímu světu [5].

Souřadný systém Báze (Base)
Tento kartézský souřadný systém je typický pro roboty KUKA. Ostatní výrobci však mají jeho obdobu, například ABB ho nazývá Workobject. Definuje jej uživatel a určuje polohu jednotlivých součástí vzhledem ke světovému souřadnému systému. V případě potřeby může jeden souřadný systém báze odkazovat i na jiný systém báze, například pokud máme nějaký přípravek umístěný na stole [5].

Souřadný systém nástroje (Tool)
Důležitým souřadným systémem je souřadný systém nástroje. V základním nastavení bývá umístěn ve středu příruby robotu a podle použitého nástroje se přesouvá obvykle do bodu na nástroji, který přichází do kontaktu se součástí. Obecně bývá značen TCP. [5] Všechny souřadné systémy můžeme vidět na obr. 4.



OBR. 4 SOUŘADNÉ SYSTÉMY ROBOTU

3.3.2 Základní pohyby robotů

Rozlišujeme tři základní pohyby, které jsou vždy definovány pro souřadný systém nástroje.

Obecný pohyb

Říká se mu také point to point pohyb. Robot se při tomto pohybu pohybuje z bodu počátečního do koncového bodu pohybu nejrychlejším možným způsobem. Toho se docílí tím, že se jednotlivé osy co nejrychleji dostanou do pozice, ve které mají být. Trajektorie pohybu není přesně definovaná.

Lineární pohyb

Robot se pohybuje přesně po přímce mezi počátečním a koncovým bodem pohybu, při čemž se pohybuje konstantní definovanou rychlostí.

Kruhový pohyb

Pohyb po kruhové dráze z počátečního do koncového bodu pohybu. Robot se opět pohybuje definovanou konstantní rychlostí.

3.3.3 Výrobci robotů

Roboty ABB

Společnost vznikla v roce 1988 sloučením firem ASEA a Brown Boveri. Zaměření společnosti je široké, od řešení pro vysoká a nízká napětí, přes pohony, řízení až po robotiku. Nabídka robotů ABB je velká, od manipulačních robotů pro velká i malá zatížení, speciální konstrukce robotů pro svařovací a lakovací aplikace, až po kolaborativního robota. Roboty ABB nesou označení IRB a dále konkrétní číslo modelu. Dále společnost dodává řídící systém IRC5 ve čtyřech variantách. V neposlední řadě v nabídce firmy najdeme různá aplikační zařízení a příslušenství jako jsou polohovadla, chapadla nebo zařízení pro lakování a svařování. Na programování a simulaci nabízí software RobotStudio [6].

Roboty KUKA

KUKA vznikla roku 1898 v Augsburgu. Postupně se od výroby plynu, přes různé svařovací aplikace propracovala až výrobě robotů a dodavatele automatizačních řešení. V roce 2016 přešla KUKA do vlastnictví čínského koncernu Midea Group. KUKA nabízí škálu robotů podobnou té od ABB. Roboty označuje KR a dodává k nim řídící systém KR C5. I KUKA nabízí k robotům další příslušenství, jako jsou polohovadla a především svařovací nářadí [7].

Roboty Fanuc

Fanuc založil v roce 1956 Dr. Seiuemon Inaba, který je spolutvůrcem číslicového řízení. Firma se soustředí převážně na robotiku a automatizaci. Kromě robotů nabízí také řídící systém pro CNC nebo řešení pro laserové, vrtací a přesné stroje. Portfolio robotů je velice podobné tomu od firmy ABB. Značení robotů se odvíjí od jejich robotů. Řízení robotů zajišťuje řídící jednotka R-30iB a nechybí další příslušenství, včetně polohovadel. Obdobně jako ABB i Fanuc nabízí vlastní software pro programování a simulaci zvaný ROBOGUIDE [8].

Další výrobci

Výrobců robotů je poměrně velké množství. Výše uvedené značky patří mezi nejpoužívanější dodavatele. Dalšími výrobci robotů pak jsou firmy jako Mitsubishi, Panasonic, Yaskawa nebo Hyundai. Mezi menší dodavatele se řadí firmy Epson, Stäubli nebo Omron. V poslední době se také začíná prosazovat firma Universal Robots se svými kolaborativními roboty za nízkou cenu.

3.4 Programování robotů

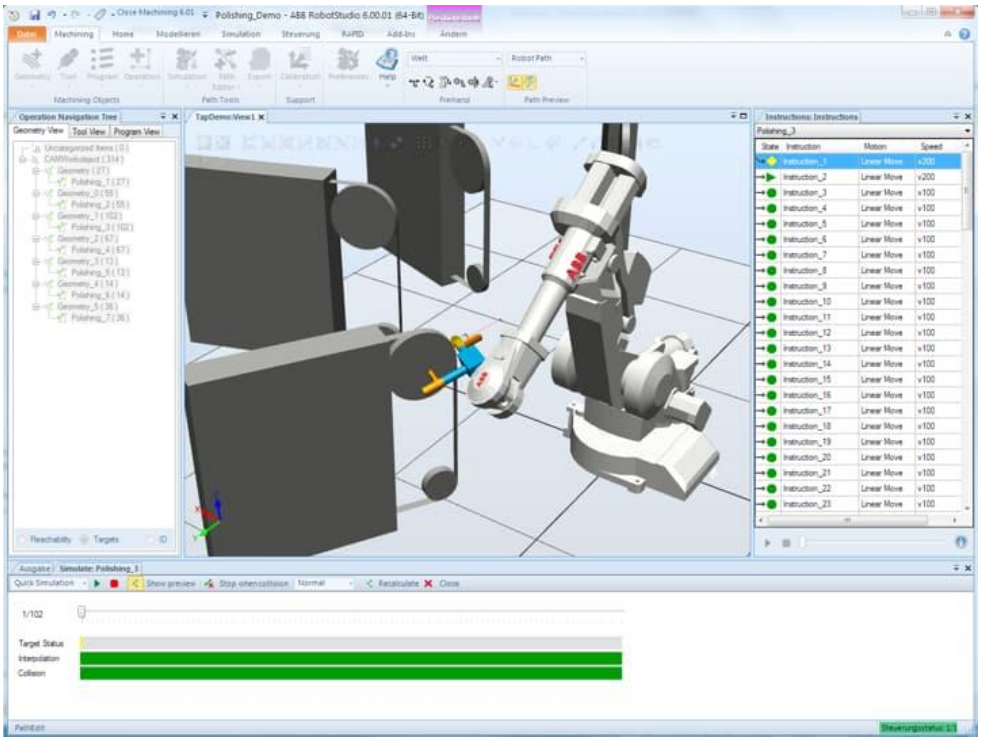
3.4.1 Způsoby programování

Programování robotů se rozděluje na dva základní způsoby, které. Prvním způsobem je takzvané online programování. Pro programování se používá ovládací panel robotu (teach pendant), který je dodávaný s řídícím systémem robotu. V takovém případě se jedná o nepřímé online programování. Pomocí pendantu programátor ovládá přímo robot, najede s ním na potřebné pozice a ty pak uloží jako pracovní body. Z nich pak tvoří trasy pro robot a navazuje na ně logiku programu. Kromě pendantu se dá využít i přímé online programování, a to vedení robotu. To se využívá například při práci s kobotem. Nevýhodou online způsobu programování je nutnost mít u sebe funkční robot, což obvykle znamená programovat vše přímo na hale. Dále pak menší přehlednost a ergonomie pendantu oproti klasickému počítači.

Druhým způsobem je offline programování. Programování probíhá na počítači bez nutnosti připojení k robotu. Nejčastěji je prováděno pomocí 3D modelu pracoviště, kde si programátor vybere body a křivky, po kterých se má robot pohybovat a nemusí na ně robot pracně navádět. Tvorba logiky programu je obdobná jako u online programování. Většinou bývá offline programování doplněno o doladění programu v režimu online. Jedná se však obvykle pouze o zprovoznění robotu a nastavení souřadných systémů [5]. Obr. 5 ukazuje programování v softwaru RobotStudio od firmy ABB [6].

3.4.2 Programovací jazyky robotů

Bohužel není používán žádný jednotný jazyk, ale každý výrobce robotů si vyvíjí svůj vlastní. Například společnost ABB má jazyk zvaný Rapid, KUKA vyvinula programovací jazyk KRL a Fanuc používá jazyk zvaný Karel. Jednotlivé jazyky jsou si však velice podobné a liší se především syntaxí a různými knihovnami. Na rozdíl od jazyků používaných pro programování PLC jsou všechny robotické jazyky textové. Většina vychází z Pascalu a jedná se o vysokoúrovňové jazyky.



OBR. 5 OFFLINE PROGRAMOVÁNÍ V ROBOTSTUDIU OD ABB

3.5 Koncové efektory robotů

Koncový efektor je zařízení, umožňující robotu provádět činnost, která je od něj požadovaná. Zároveň se jedná o jednu z mála součástí robotů, která je obvykle navrhována nově pro danou aplikaci. Existují sice některá řešení pro častější úkony a požadavky, většinou je však potřeba vytvořit koncový efektor na míru. Konstruktéři v takovém případě nevytvářejí všechny prvky efektoru, ale využívají různých částečných řešení od dodavatelů, soustředících se na výrobu prvků pro konkrétní aplikaci. Hlavním úkolem konstruktéra při návrhu koncového efektoru tedy je zkombinovat řešení různých dodavatelů a zajistit dostatečné síly v akčních členech efektorů.

Výjimkou nejsou výměnné nebo dvojité efektory. První typ se používá, pokud má robot zajišťovat více odlišných operací nebo manipulovat s různě velkými či tvarově odlišnými předměty. Druhý typ je pak vhodný například při manipulaci s více předměty najednou pro ušetření času. Koncový efektor pak může obsahovat jak dvě stejné, tak i různé jednotky.

3.5.1 Grippery

Gripper nebo také Greifer je koncový efektor určený pro uchopení součásti. Uchopení u tohoto typu efektoru probíhá sevřením čelistí, podobně jako u sklíčidla. Grippery bývají obvykle dvou čelisťové nebo tří čelisťové, ale nejsou výjimkou i další řešení. Kromě počtu čelistí se dělí také podle pohonu efektoru. Nejčastěji bývá síla v čelisti vytvářena buď pomocí stlačeného vzduchu, poté mluvíme o pneumatickém gripperu, nebo pomocí servomotoru a převodovky, která kromě úpravy momentu zajišťuje také převod rotačního pohybu hřídele serva na lineární pohyb čelistí.

Jedním z největších dodavatelů technologií pro uchopovací systémy je firma Schunk. V jejím sortimentu jsou k dispozici paralelní, středící, úhlová, magnetická, rotační a další chapadla. Kromě samotných chapadel, jejichž ukázka je na obr. 6, se firma soustředí i na další příslušenství robotů, jako jsou mezikusy sloužící pro připevnění jejích chapadel k přírubám robotů od určitých výrobců, upínací techniku pro obráběcí centra a v neposlední řadě na zakázková řešení [9].

OBR. 6 PŘÍKLADY CHAPADEL FIRMY SCHUNK

3.5.2 Vakuové přísavky

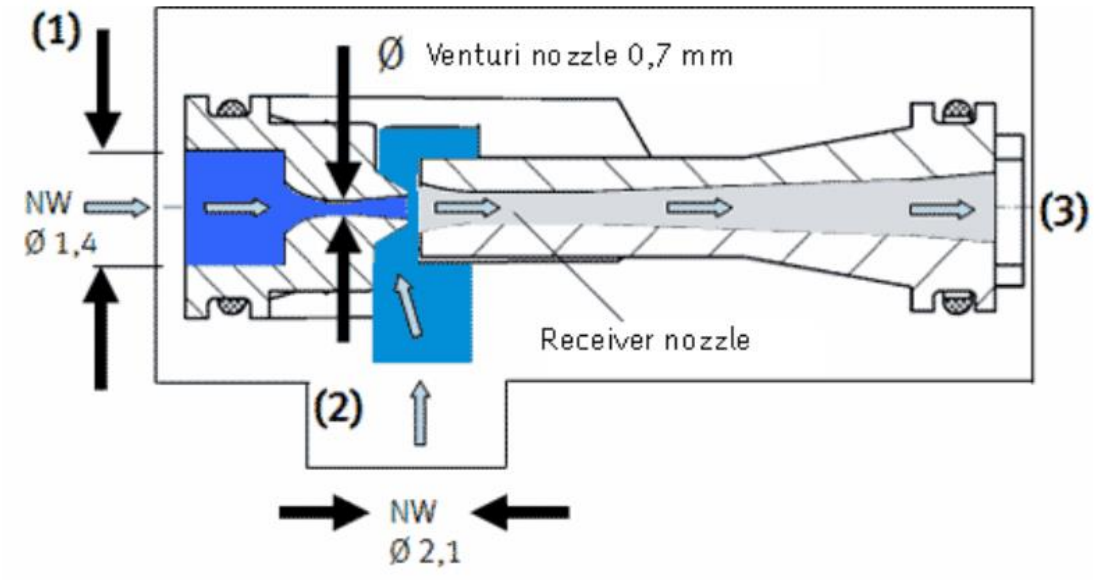
Vakuové přísavky fungují na principu odsávání vzduchu ze zatěsněného prostoru. Přísavky mohou mít různý tvar, nejčastěji se však používají přísavky kruhové. Podle tvrdosti a členitosti dotykové plochy součásti jsou pak voleny materiály přísavek. V případě, že může mít předmět poddajný, používají se přísavky s pružinou, která zajistí vždy správné dosednutí přísavky na povrch. Obecně se vakuové přísavky používají pro rovné povrchy kvůli jednoduššímu utěsnění, nicméně výrobci jsou schopni dodat řešení i pro mírně členité povrchy.

Kromě samotné přísavky, která zajišťuje kontakt s předmětem je potřeba ke správné funkci také pneumatický obvod. Kromě přivedení stlačeného vzduchu, ať už centrálním rozvodem vzduchu v hale nebo samostatného kompresoru, jednotky pro úpravu vzduchu a veškerého potřebného vedení hraje důležitou roli ejektor. Kompresor totiž dodává vzduch stlačený, v hadici tedy vzniká přetlak. Pro vysátí vzduchu z prostoru přísavky přiložené k objektu manipulace je však potřeba podtlak. Ejektor samotný funguje na základě Venturiho efektu, kdy vzduch vstupuje (1) do sací dýzy, která je buď klasicky se zužující nebo Lavalova. [10] Díky zúžení v dýze se zvýší rychlost proudění vzduchu na úroveň nadzvukové rychlosti. Po výstupu z dýzy vzduch zvětší svůj objem a proudí tryskou přijímače na konec ejektoru (3), kde vystupuje směs vstupního a nasátého vzduchu do tlumiče hluku. Při tom se v komoře mezi dýzou tryskou přijímače tvoří vakuum, které způsobuje nasávání vzduchu ze sacího otvoru (2). Funkci ejektoru popisuje obr. 7 [11].

Mezi největší dodavatele vakuových přísavek a pneumatiky obecně patří japonská společnost SMC. Dalším velkým hráčem na poli pneumatiky je německá firma Festo, která kromě pneumatických řešení dodává také hydraulické výrobky.

3.5.3 Speciální efektory

Kromě výše zmíněných se můžeme setkat také se efektory pro konkrétní aplikace. Nejčastěji se jedná o svařovací hlavice nebo lakovací trysky, ale výjimkou nejsou také efektory obráběcí nebo měřící. Tyto efektory však bývají většinou dodávány jako hotové řešení externím dodavatelem a nebývají konstruovány automatizační firmou, která je pak pouze zakomponuje do pracoviště.



OBR. 7 PRINCIP EJEKTORU

3.6 Popis PLC

PLC je zkratka pro průmyslový kontroler, používaný především pro automatizaci procesů. Jedná se o uzavřený systém, podobný počítačům, pro který výrobce většinou vyvíjí vlastní operační systém a drivery. Používají se však i tak zvané IPC, které běží na operačním systému externího dodavatele, většinou se jedná o Windows od firmy Microsoft.

Průmyslové kontrolery se od klasických desktopových počítačů nebo notebooků liší v několika zásadních bodech. Jedním z nich jsou použité součástky. Typy součástek (CPU, operační paměť atd.) jsou stejné. Nicméně zatímco v novém notebooku je většinou použit nejnovější typ procesu a nový, progresivní hardware, u PLC je tomu jinak. Hardware použitý pro PLC musí být vyzkoušený a ověřený, ideálně tím, že byl několik let nasazený v jiné aplikaci. Je zde totiž požadován bezchybný chod často v nepřetržitém třísměnném provozu po doby několika let.

Výměna PLC totiž vždy skýtá hrozbu komplikací při opětovném zapojení všech periferií, nahrání programu a znovu spuštění. Při tvorbě automatizovaného pracoviště či jednoúčelového stroje je nutné před spuštěním vše vyladit a odstranit veškeré chyby. Tento proces je často časově náročný. Pokud by PLC sloužilo pouze pět let, jako většina hardwaru používaného v konzumní elektronice, hrozilo by, že po jeho výměně se tento zdlouhavý proces ladění stroje bude muset opakovat. To si však zákazník nepřeje, jelikož to pro něj znamená ztráty, když nemůže vyrábět. Z tohoto důvodu je při výběru hardwaru PLC kladen důraz především na spolehlivost a až v druhé řadě na výkon. Ten je však také důležitý. Používá se tedy často hardware, který je odzkoušený na konzumní elektronice nebo serverech a s případnými drobnými úpravami je pak implementován právě do PLC. Jelikož je kvůli tomuto omezení počet dostupných součástek menší než při návrhu konzumní elektroniky, PLC bývají při stejném výpočetním výkonu mnohem dražší než klasické počítače.

3.6.1 Druhy PLC

Mikro PLC

PLC většinou rostou na velikosti, pokud po nich chceme větší výkon, více vstupů či výstupů nebo se musí připojit k více zařízením. Většinou jsou tak kvůli stardardizaci jednotlivých řad i kompaktní PLC poměrně velká. S postupnou rostoucí automatizací výrobních provozů a linek postupně začíná většina firem bojovat s omezeným prostorem, a tak vzniká stále větší tlak i na co nejmenší zástavbové rozměry rozvaděčů, skříní i samotných PLC, která jsou často s frekvenčními měniči hlavními prvky zabírajícími místo.

Mimo velké stroje a linky, které vyžadují u řídících jednotek větší počet vstupů a výstupů, se často objevují i malé aplikace v podobě nutnosti odřídit pohyb jednoho samostatného dopravníku nebo pár metrů potrubní cesty s několika pneumatickými ventily. V takovém případě řešení vystačí jen s několika vstupy a výstupy a případně jednou sběrnicí, například pro ovládání rychlosti řídící jednotky pohonu.

Výhodou těchto PLC je tedy jejich již zmíněná velikost a nízká cena. Největší nevýhodou pak jsou především omezené možnosti vstupů a výstupů. Dalším limitujícím parametrem bývá výkon těchto zařízení.

Kompaktní PLC

Jedná se o typ PLC, kdy je v jednom modulu integrováno nejenom řízení, ale také vstupy a výstupy, případně další prvky, potřebné k řízení. Počty vstupů i výstupů jsou dány výrobce a většinou je nelze rozšiřovat dalšími moduly [12].

Kompaktní PLC bývají levnější než modulární provedení. Další výhodou bývá přístup k jednotlivým vstupům a výstupům, jelikož je vše u sebe a nemusí se řešit žádné sběrnice. To může mí také vliv na dobu cyklu (takt). Nevýhodou je pak samozřejmě omezený počet vstupů a výstupů, i když jsou i systémy, které umožňují rozšíření [12].

Modulární PLC

Jak již vyplývá z názvu, modulární PLC jsou tvořeny jednotlivými moduly. Jedním z nich musí být samozřejmě CPU, neboli řídící modul. Dále je většinou použit jeden nebo více modulů pro vstupy a výstupy. Ty mohou být analogové nebo digitální. Lze také připojit různé komunikační moduly pro připojení k internetu (s datovým konektorem nebo wifi modulem), dalších PLC či jiných periferií pracoviště, jako jsou například roboty či jiné řídící systémy.

Modulární PLC kromě zmíněných modulů tvoří i tzv. „Case“. Jedná se o jakousi obdobu PC krabice nebo serverových racků. Není to nic jiného než kostra, do které se jednotlivé moduly montují, většinou na DIN lištu. Podle požadavků zákazníka a především prostředí, ve kterém má PLC pracovat, pak může mít různou IP odolnost.

Kromě modulů a kostry je u modulárních PLC nutno počítat také s kabeláží. Jednotlivé moduly jsou mezi sebou a CPU propojeny datovými sběrnicemi, které jim umožňují spolu komunikovat. Příklad celého rozvaděče modulárního PLC je na obr. 8 [13].

OBR. 8 PŘÍKLAD PLC ROZVADĚČE S MODULÁRNÍM PLC EATON

3.6.2 Výrobci PLC

PLC Siemens

Siemens je německý výrobce a patří mezi největší dodavatele na poli PLC a automatizace, především v Evropě. Modulární PLC v nabídce Siemens momentálně zastupuje PLC SIMATIC. Všechna PLC v nabídce firmy Siemens se programují ve vývojovém prostředí zvaném TIA portal (Totally Integrated Automation Portal). Jedinou výjimkou je základní model LOGO, který se programuje v prostředí LOGO Soft. TIA portal umožňuje kromě programování také simulaci, k vizualizaci je však potřeba originální displej [14].

PLC Beckhoff

Společnost Beckhoff Automation GmbH & Co. KG je výrobcem automatizační technologie se sídlem ve Verlu ve východním Westfálsku a je součástí skupiny Beckhoff Group. Modulární PC jsou obsaženy v řadě CX. Programování se prování přes software Twincat 3, případně starší verzi Twincat 2. Ve vývojovém prostředí lze provést veškeré simulace, včetně vizualizací. Vývojové prostředí TwinCAT 3 lze stáhnout z internetu zcela zdarma. Uživatel platí jen run-time licence na počítačích, které jsou nasazeny v reálných aplikacích. Pro vyzkoušení a odladění vyvíjené aplikace lze zdarma využít zkušební licenci pro run-time, která je plně funkční po dobu 7 dní [15].

PLC Allen Bradley

Allen Bradley je zavedený americký výrobce PLC, včetně modulárních. Značka patří firmě Rockwell Automation a byla založena roku 1903. Zástupcem modulárních PLC této značky je řada ControlLogix, kterou můžeme vidět na obr. 9. K návrhu a konfiguraci systému ControlLogix se používá Studio 5000 Logix Designer od společnosti. Řadiče a moduly, které se při návrhu vyberou spolu s konfigurací sítě určují, jaké další softwarové balíčky budou potřeba pro konfiguraci a programování systému. Další požadovaný software může zahrnovat konfigurační software RSNetWorx a komunikační software RSLinx [16].

OBR. 9 ALLEN BRADLEY PLC CONTROLLOGIX

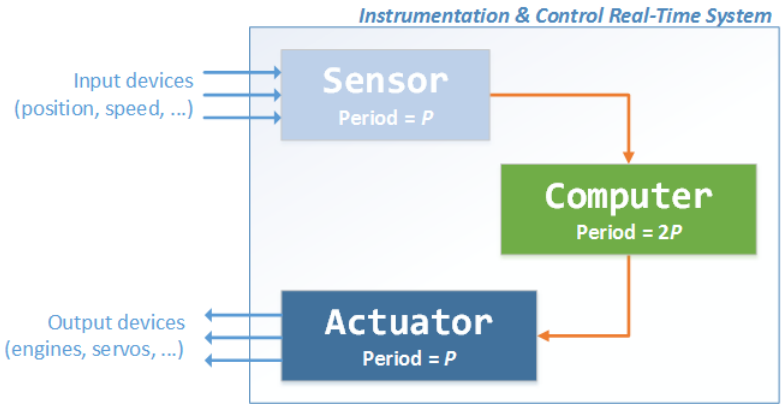
Další výrobci

Firem, které se zabývají výrobou PLC je celá řada a v této práci bylo uvedeno jen několik příkladů výrobců, jejichž výrobky jsou nejčastěji používané v průmyslu v Evropě a Americe. Mezi další výrobce se však řadí např. Bosch, ABB, Omron, Eaton a další.

3.6.3 Operační systém PLC

Dalším bodem, ve kterém se PLC liší od klasických PC je operační systém. Můžeme se sice setkat s výrobci, kteří používají řídící systém od zavedených softwarových firem, jako je například Windows (použitý firmou Beckhoff) nebo Linux (použitý firmou Eaton). Jedná se však o speciální verze těchto systémů, které nejsou běžně k dostání. Základní vlastností operačního systému PLC je, že musí být tzv. „Real time“. Zatímco běžné operační systémy vykonávají jeden úkol za druhým a k druhému nepřejdou, dokud nedodělají první, real time systémy pracují v taktech. To znamená, že všechny úkoly a výpočty musejí proběhnout v určitém časovém intervalu (tzv. taktu) a pokud se tak nestane, systém se přepne do chybového stavu. Tento způsob provádění úkolů je nutný kvůli bezpečnosti. Jednoduché schéma principu real time operačního systému je na obr. 10 [17].

Zatímco u běžného PC nám většinou nevadí si chvíli počkat na vyhotovení všech úkolů, ve výrobě je nepřípustné, aby se pohon vypnul tři vteřiny po signálu z koncového senzoru jen proto, že systém ještě nedokončil předchozí výpočet. Navíc při složitějších výrobních a montážních operacích, kdy je potřeba koordinovat několik pohybů zároveň, by docházelo k výrobním nepřesnostem kvůli špatnému načasování nebo minimálně ke zpoždění výroby, což by znamenalo finanční ztrátu.



OBR. 10 ZNÁZORNĚNÍ REAL TIME SYSTÉMU

3.7 Programování PLC

PLC se programují ve vývojových prostředích podobných těm, které se používají v klasickém programování. Rozdíl je především v tom, že zatímco při klasickém programování bývá vývojové prostředí spojeno především s jazykem (například PyCharm pro Python), případně podporuje i další používanější typy jazyků, u vývojových prostředí pro PLC to je trochu jinak. Dodává je totiž každý výrobce zvlášť ke svému PLC. Výhodou tohoto přístupu je snadné nahrání programů do řídících modulů a práce s I/O vstupy. Nevýhodou však je především cena. PLC vývojová prostředí jsou totiž, na rozdíl od těch klasických, zpoplatněna. Dalším nedostatkem je uzavřenost těchto prostředí, takže se hůře dělají různé úpravy pro zjednodušení práce. Některá kompaktní PLC pak umožňují jednoduché úkony nastavit pomocí hardwarových tlačítek umístěných na čelní straně.

3.7.1 Programovací jazyky PLC

PLC se většinou neprogramují pomocí standardních programovacích jazyků. Jedním důvodem je real-time operační software a jeho náležitosti, druhým je pak potřeba určité standardizace. Běžné programovací jazyky se velice rychle vyvíjejí a jejich frameworky se mění. To by byl v průmyslových aplikacích problém, jelikož je někdy potřeba upravit program na stroji, který je několik let starý a neumožňuje použití nejnovějšího firmwaru. Z těchto důvodů byly vytvořeny určitě standardy, mezi něž patří programovací jazyky ST, LDR, FB a další, definované normou IEC 61131-3 [18].

ST (Structured Text)

ST je jazyk založený na syntaxi Pascalu. Jedná se o jediný textový jazyk používaný pro programování PLC. Jelikož vychází z klasického programovacího jazyku, je nejčastěji používán programátory, kteří do oboru přešli právě ze světa klasických počítačů. Jeho jedinou nevýhodou oproti ostatním jazykům je menší přehlednost. To však platí pouze pro menší projekty, kde jsou ostatní, grafické, jazyky přehlednější. U větších projektů bývá často naopak přehlednější, jelikož zabírá méně místa. Jeho největší výhodou je možnost verzování, což u ostatních jazyků určených pro PLC možné není.

LDR (Ladder Diagram)

LDR je grafický jazyk, který vychází z reléové logiky. Když se začaly v řízení strojů prosazovat mikročipy na úkor relé, nebyl dostatek programátorů schopných tvořit programy v klasických textových jazycích, tak jako na PC. Byl však naopak přebytek elektrotechniků, schopných vytvořit logiku stroje pomocí relé. Proto vznikl grafický jazyk, založený na logice tvorby reléových schémat. Logika se v tomto jazyce tedy vytvoří podobně jako se dříve tvořila schémata pro reléová zapojení, tedy v grafické formě. Grafické zobrazení pak je převedeno na klasický kód, se kterým pak už pracuje kompilátor. Tento jazyk je tedy jednoduchý na naučení pro elektrotechniky a zároveň docela prostorově úsporný. Jeho nevýhodou je nemožnost verzování.

FB (Function Block Diagram)

FB je další grafický jazyk. Jeho hlavním účelem je přehlednost a relativní jednoduchost, především oproti jazyku ST. Programování se provádí pomocí vkládání jednotlivých funkčních bloků, které obsahují potřebné vstupy a výstupy. Daní za přehlednost je velká prostorová náročnost programu, takže při větších projektech se v programu může hůře pohybovat. Další nevýhodou je nemožnost verzování, jelikož jednotlivé bloky jsou tvoře XML soubory a při každém posunutí bloku v programu se změní jemu přirazený XML soubor.

3.7.2 Datové typy

Datové typy jsou důležitou součástí programování a u programování PLC je na ně potřeba hledět ještě více než u klasických PC. Složitější datový typ totiž potřebuje větší velikost paměti na zápis, proto se při použití zbytečně velkých datových typů může stát, že zvolená konfigurace PLC nebude stíhat program vyhodnotit v potřebném časovém cyklu. To má za následek nutnost použití vyšší řady PLC, což zvedá výslednou cenu stroje.

Pro jednoduchost lze datové typy rozdělit do několika kategorií. První jsou základní datové typy. Jedná se o jednoduché datové typy s velikostí do 32 bitů a mají pevnou délku. Souhrn nejpoužívanějších základních typů ukazuje tab. 1 [12].

TAB 1 ZÁKLADNÍ DATOVÉ TYPY

DATOVÝ TYP	VELIKOST (BIT)	POPIS
BOOL	**1**	Boolean
BYTE (Byte)	**8**	hexadecimální číslo / BCD
WORD (Word)	**16**	binární zobrazení / hexadecimální číslo / BCD
DWORD (Double Word)	**32**	binární zobrazení / hexadecimální číslo / BCD
INT (Integer)	**16**	celé desítkové číslo se znaménkem
DINT (Double Integer)	**32**	celé desítkové číslo se znaménkem
REAL (Reálné číslo)	**32**	číslo s plovoucí desetinnou čárkou
TIME (IEC time)	**32**	čas, integer se znaménkem, rozlišení 1 ms
DATE (IEC date)	**16**	datum, rozlišení 1 den
TIME OF DAY (time)	**32**	čas v rozsahu 1 den, rozlišení 1 ms
CHAR	**8**	znak ASCII

Další kategorií jsou komplexní datové typy. Ty jsou větší než 32 bitů a mohou, ale také nemusí mít pevnou délku. Nejčastější datovým typem s proměnnou délkou je pole (array). Komplexní datové typy ukazuje tab. 2 [12].

TAB 2 KOMPLEXNÍ DATOVÉ TYPY

TYP	VELIKOST	POPIS
DATE AND TIME	64 bit	datum a čas
STRING	254 Byte	textový řetězec
ARRAY	-	více rozměrová oblast dat (vektor, matice,...) jednoho typu dat
STRUCT	-	strukturovaná oblast jednoho nebo různých typů dat

Třetí skupinou jsou datové typy pro formální parametry, které doplňují předchozí dvě kategorie. Často se jedná o formální parametry, které definují přenos mezi programovými bloky. Tyto datové typy popisuje tab. 3 [12].

TAB 3 DATOVÉ TYPY PRO FORMÁLNÍ PARAMETRY

DATOVÝ TYP	VELIKOST (BIT)	POPIS
TIMER	**2 Byte**	číslo časovače (integer)
COUNTER	**2 Byte**	číslo čítače (integer)
BLOCK	**2 Byte**	číslo bloku (programový nebo datový) (integer)
POINTER	**6 Byte**	identifikátor proměnné a adresy, odkazuje na adresu proměnné
ANY	**10 Byte**	pro použití, kdy je datový typ aktuálního parametru neznámý nebo lze použít
VARIANT	**PROMĚNNÁ**	ukazatel na proměnné různých typů

Datové typy z předchozích tabulek se dají rozdělit také podle použití. Běžně se totiž nepoužívají všechny datové typy pro zápis jakýchkoli informací. Je například zbytečné použít typ integer pro zápis hodnoty 0 nebo 1 (true/false). V takovém případě je mnohem vhodnější použití typu BOOL, jelikož zapsaní hodnoty pak potřebuje mnohem menší velikost paměti. Rozdělení podle použití ukazuje tab. 4 [12].

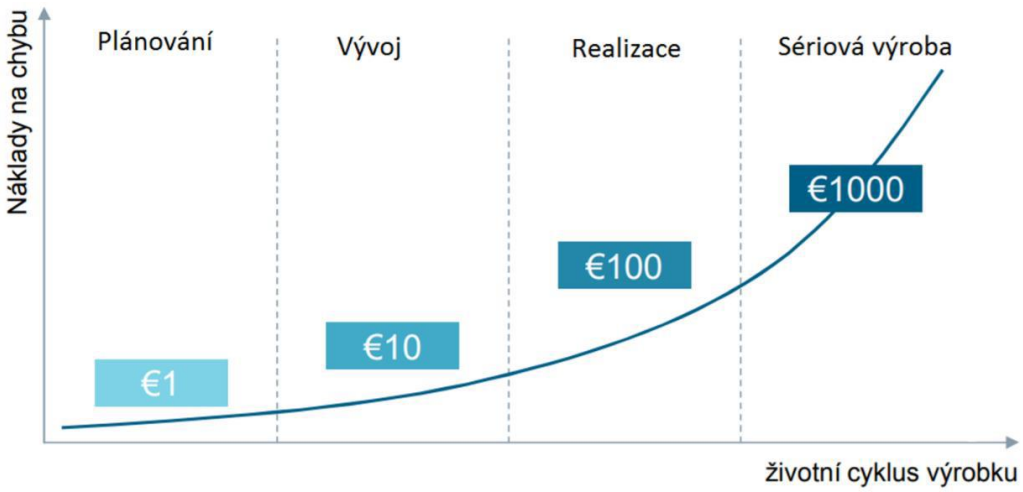
TAB 4 DATOVÉ TYPY PODLE POUŽITÍ

POUŽITÍ	TYP
BIT(Y)	BIT, NIBL, WORD, DOUBLE WORD, ...
ČÍSLO	INTEGER, DOUBLE INTEGER, REAL, LONG REAL
ZNAK(Y)	CHAR, STRING, ...
ČAS, DATUM	TIME, DATE, TIME OF DAY, DATE AND TIME
UŽIVATELSKÉ	ARRAY, STRUCT
SPECIÁLNÍ	ANY, POINTER, VARIANT

3.8 Virtuální zprovoznění

Po dohodnutí parametrů výrobní linky mezi zákazníkem a dodavatelem, práce na výrobní lince standardně začíná na konstrukci. Konstruktér navrhne vhodné rozmístění jednotlivých prvků pracoviště, vybere vhodná řešení od různých výrobců, navrhne koncové efektory, pneumatická schémata a podobně. Zároveň zodpovídá za mechanickou funkčnost a často také bezpečnost pracoviště, tedy aby nikde nedocházelo ke kolizím a všechny prostory byly vhodně oploceny či jinak odděleny od okolí. Když je vytvořený konstrukční návrh pracoviště, navrhne se elektro projekce, kdy je navržen rozvaděč, kabeláž a vybrány jednotlivé elektrotechnické prvky jako motory nebo jištění. Poté jsou nahrubo vytvořeny programy pro PLC a roboty. Následně je linka vyrobena, přepravena k zákazníkovi, sestavena a zprovozněna. Při zprovozňování linky se ladí programy, často se dodělává jejich převážná část, vše se zkouší na hardwaru a programátor často na pracovišti u zákazníka stráví několik týdnů. Pokud se přijde v této fázi zprovozňování a testování na nějaký problém s návrhem pracoviště, je nutné navrhnout, vyrobit, přepravit a nainstalovat nový díl, což stojí nemalé peníze.

Virtuální zprovoznění slouží k zjednodušení a především zlevnění testovací fáze návrhu linky a také k urychlení programování. Testování na zprovozněném hardwaru je samozřejmě potřeba i při použití virtuálního zprovoznění, avšak je minimalizována pravděpodobnost vzniku chyby nebo opomenutí. Dosahuje se toho za pomoci simulačního softwaru, do kterého se z CAD softwaru přenese pracoviště navržené konstruktérem, ideálně doplněné o prvky z elektro projekce. Programátor si v tomto softwaru může navrhnout trasy pro pohyby robotů, simulovat pohyby robotů, dopravníků a ostatních prvků linky a v neposlední řadě také simulovat tok materiálu, se kterým linka pracuje. Když má vše navrženo, může si vygenerovat programy, ty si případně upravit podle potřeby a simulovat jejich úpravy opět ve stejném softwaru. Kromě usnadnění programovací části návrhu linky spočívá největší přínos simulačního softwaru právě v ověření různých scénářů, kolizí a potenciálních problémů, které by mohly na lince vzniknout ještě před začátkem výroby a montáží linky. Tento postup může potenciálně ušetřit nemalé náklady, obzvlášť pokud odhalí problém, na který by se jinak přišlo až při spuštění výroby na lince. Náklady na odstranění chyb při vývoji ukazuje obr. 11 [19].



OBR. 11 NÁKLADY NA ODSTRANĚNÍ CHYBY PŘI VÝVOJI

4 POPIS FUNKCE A ROZMÍSTĚNÍ PRACOVIŠTĚ

4.1 Operace na pracovišti

Na vstup jedoucího dopravníku je umístěna součást. Když součást dojede k prvnímu snímači, který je umístěný pod kamerou, kamera udělá snímek dopravníku včetně součásti. Vyhodnocení snímku obsahuje určení druhu součásti, případně upozornění, pokud součást nebude nalezena, dále zjištění X a Y souřadnic dohodnutého bodu na součásti, natočení součásti oproti zvolenému souřadnému systému a kontrolu shodného či neshodného dílu. Zjištěné parametry jsou pak poslány do řídícího PLC. Toto vyhodnocení není součástí této práce a je zde uvedeno pouze pro přiblížení funkce celého pracoviště.

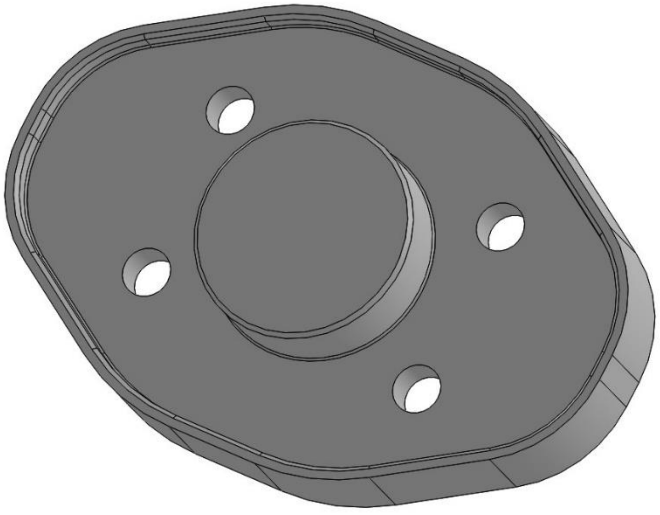
Při detekci součásti na začátku dopravníku je kromě pořízení snímku také poslán signál do PLC, který spustí timer. Ten slouží pro vyhodnocení chybového stavu, pokud by se součást na dopravníku zasekla nebo nastal jiný problém, kvůli kterému by součást nedojela na konec dopravníku.

Na konci dopravníku je umístěn druhý snímač, který při detekci součásti zastaví dopravník. Zároveň je zastaven timer, který byl spuštěný snímačem na začátku dopravníku. Proběhne přečtení zprávy z kamery a robot najede na zjištěnou pozici součásti, odebere ji z dopravníku a umístí ji na stůl. Po navrácení robotu do HOME pozice je opět spuštěn dopravník a celý proces se opakuje, dokud nedojdou díly na vstupu, není vyčerpán prostor na odkládacím stolu nebo nenastane chybový stav.

Robotické operace prováděné na pracovišti tedy nejsou samy o sobě složité, jelikož se jedná o klasické pick and place operace. Náročnější je sladění všech prvků řízení a převedení pracoviště do softwaru pro virtuální zprovoznění.

4.2 Objekty pro manipulaci

Prvním objektem je plastová zátka. Pro detekci kamerou ani pro uchycení robotem se nejedná o složitý díl. Materiál dílu je plast, celkové rozměry jsou 150 x 100 x 30 mm a hmotnost je 248 g. Plocha pro přísavku koncového efektoru je kruhového tvaru o průměru 48 mm. Model dílu na obr. 12.



OBR. 12 ZÁTKA

Dalším objektem je opakovatelně použitelný plastový obal pro dětské nápoje Papoo Original. Jeho nevýhodou při sejmutí robotem z dopravníku je jeho proměnná výška. Tento problém se dá řešit buď hardwarově, použitím přísavky s odpružením, nebo softwarově v programu robotu. Základní rozměry jsou 170 x 100 mm, při čemž tloušťka obalu je kolem 1 mm. Hmotnost obalu je 10 g, což je zanedbatelná hodnota. Obal je vyfocen na obr. 13.

OBR. 13 PLASTOVÝ OBAL

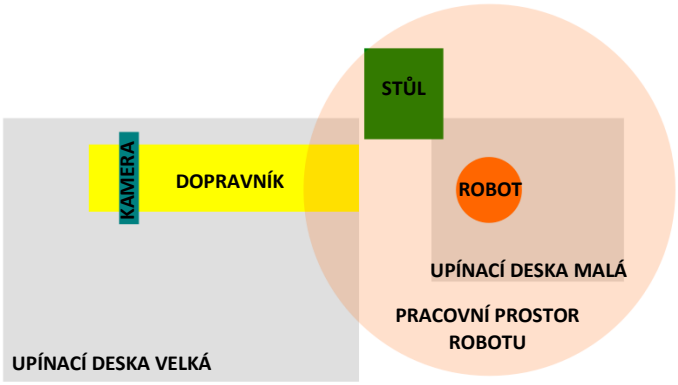
Posledním objektem je série úchytů pro vedení trubek. Jedná se o ocelové kostky se závitem vystupujícím ze jedné strany dírou skrz. V případě, že budou kostky vloženy na dopravník touto dírou vzhůru, nelze je robotem uchopit. Vnější rozměry kostek jsou 45 mm, 40 mm a 30 mm, přičemž průměry ploch pro přísavku jsou 40 mm, 36 mm a 27 mm. Hmotnosti kostek pak jsou 362 g, 219 g a 111 g. Kostky jsou vyfoceny na obr. 14.

OBR. 14 KOSTKY

4.3 Možná řešení rozmístění

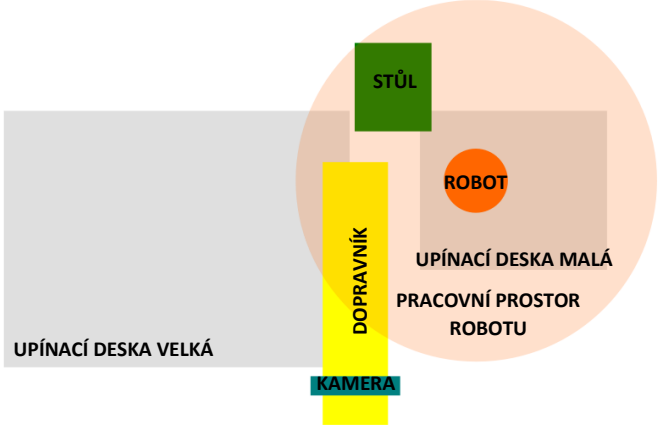
Rozmístění jednotlivých prvků na pracovišti je do jisté míry dáno již připravenými upínacími deskami a na nich namontovanými roboty. Na pracovišti se kromě robotu, dopravníku a stolu, použitých pro účely kontroly a manipulace se součástmi, nachází také druhý robot a druhý stůl. Tyto dva objekty jsou z layoutů vynechány, avšak jsou zde zmíněny jako vysvětlení přebytečného místa v layoutech.

První možnou variantou layoutu je umístění robotu na menší upínací desku, přesněji přibližně na střed její kratší strany. Díky tomuto umístění má robot dosah i da druhou upínací desku. Na té se nachází dopravník, který svým výstupem zarovnán s upínací deskou. Na vstupu dopravníku pak je umístěn stojan s kamerou. Stojan je napevno přimontován k dopravníku. Posledním prvkem na pracovišti je stůl, na který robot umístí součásti poté, co je odebere z dopravníku. Tento stůl je umístěn vedle robotu a jednou nohou je přikotven k menší upínací desce. Rozmístění jednotlivých prvků na pracovišti popisuje obr. 15. Toto rozmístění odpovídá původnímu rozmístění všech strojů na pracovišti, takže jeho výhodou je menší práce při úpravě pracoviště. Další výhodou je možnost rozšíření pracoviště o prvek přimontovaný k menší desce, který by se díky tomu nacházel v dosahu robotu.



OBR. 15 LAYOUT PRACOVIŠTĚ - VARIANTA 1

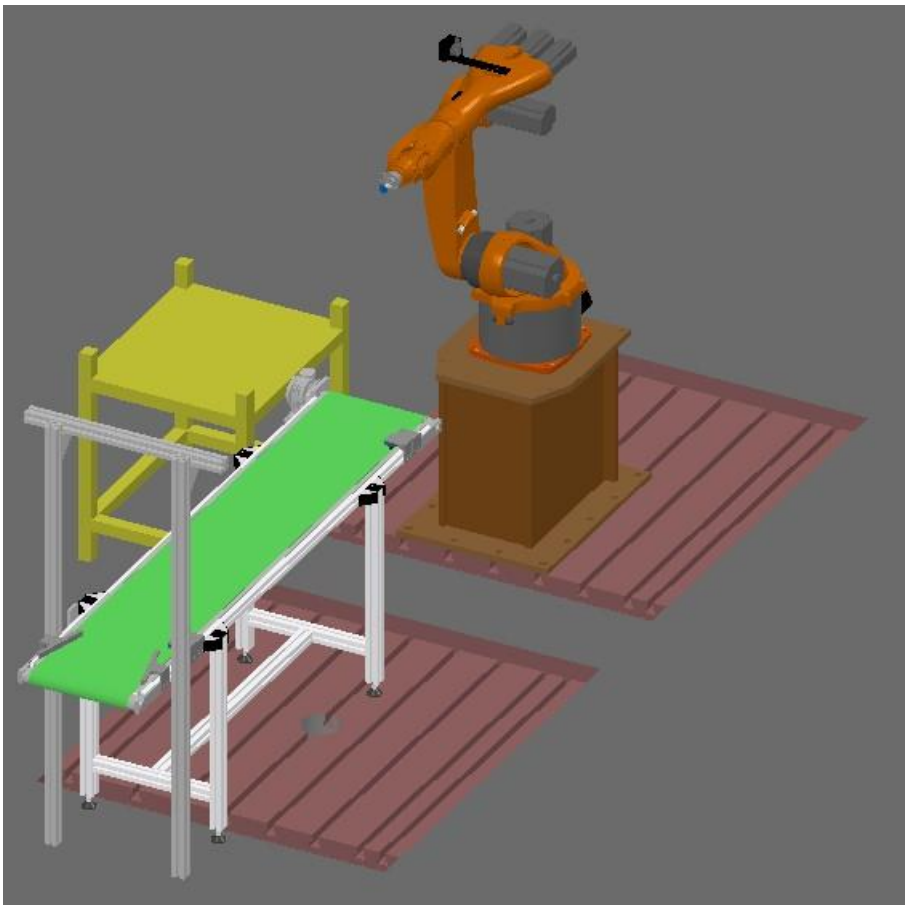
Druhá varianta je rozmístěním prvků velice podobná první variantě. Robot je umístěn na stejném místě na menší upínací desce. Odkládací stůl je také na stejném místě, vedle robotu a připevněn k menší upínací desce. Dopravník je však v tomto případě natočen o devadesát stupňů a umístěn podélně na straně velké upínací desky, která sousedí s menší deskou. Rozmístění strojů ukazuje obr. 16. Tento layout má výhodu v kompaktnosti pracoviště, jelikož téměř celá větší upínací deska je k dispozici jiným činnostem. Nevýhodou je pak práce při změně rozmístění prvků na pracovišti.



OBR. 16 LAYOUT PRACOVIŠTĚ - VARIANTA 2

4.4 Finální layout

Jako finální layout pracoviště byla vybrána první varianta. Hlavním důvodem pro volbu této varianty byl fakt, že nevyžaduje přesouvání prvků po pracovišti, prvky budou pouze upnuty k upínacím deskám. Kromě zmíněného důvodu hrála roli při rozhodování i využitelnost dopravníku druhým robotem, umístěným na velké upínací desce. Na pracovišti totiž budou probíhat i další úkony, kterým by rozmístění strojů podle druhé varianty nevyhovovalo. Finální varianta je zobrazena na obr. 17, tentokrát jako 3D model vytvořený v programu Solidworks. Pracoviště je zde již zaměřené podle skutečného rozmístění na hale. Na obrázku chybí pouze kamera.



OBR. 17 MODEL PRACOVIŠTĚ

5 POPIS POUŽITÝCH STROJŮ A ŘÍZENÍ

Většina použitých strojů a řízení na pracovišti byla dána ještě před návrhem pracoviště. Jedná se konkrétně o robot s řídícím systémem, PLC a dopravník. Tyto komponenty by se tedy daly optimalizovat a v případě návrhu pracoviště bez počátečních vstupů by některé tyto komponenty byly voleny jinak. V rámci projektu pak byly vybrány zbylé potřebné komponenty.

5.1 Dopravník

Pro přepravu dílců od kamery k robotu byl na pracovišti použit dopravník od dodavatele ALUTEC K&K. Dopravník je osazen asynchronním motorem BN 63B 4 od firmy Bonfiglioli se šnekovou převodovkou VF30 A P63 B14 [20]. Pohon je řízen frekvenčním měničem Commander SK od firmy Emerson [20]. Základní parametry dopravníku popisuje tab. 6. Pro potřeby pracoviště by však postačoval menší dopravník, než který byl použit. Model dopravníku je na obr. 18.

TAB 5 PARAMETRY DOPRAVÍKU

PARAMETR	HODNOTA
Délka	2000 mm
Šířka	350 mm
Výška	800 mm
Hmotnost	55 kg
Příkon	180 W

OBR. 18 MODEL DOPRAVNIKU

5.2 Robot KUKA

Robot, použitý na pracovišti je KUKA KR 5 arc. Primární určení robotu je pro svařovací operace, to však nevylučuje použití pro manipulaci. Kinematikou se jedná o klasický šestiosý robot. Svařovací roboty se pak vyznačují kompaktnějším designem, aby byly schopny pracovat ve stísněnějších prostorech než klasické roboty. Konkrétní hodnoty parametrů robotu popisuje tab. 5. [21] Model robotu je vyobrazen na obr. 19.

TAB 6 PARAMETRY ROBOTU KUKA KR 5 ARC

Vlastnost	Hodnota
Maximální dosah	1 411 mm
Jmenovité užitečné zatížení	5 kg
Maximální celkové zatížení	37 kg
Počet os	6
Montážní pozice	Podlaha, strop
Opakovatelnost polohování	±0.04 mm
Kontroler	KR C2 edition2005
Hmotnost (bez kontroleru)	127 kg
Teploty za chodu	+10 °C to +55 °C
Třída ochrany	IP 54
Základna	324 mm x 324 mm
Přípojka	7.3 kVA
Zvuková hladina	< 75 dB

OBR. 19 KUKA KR 5 ARC

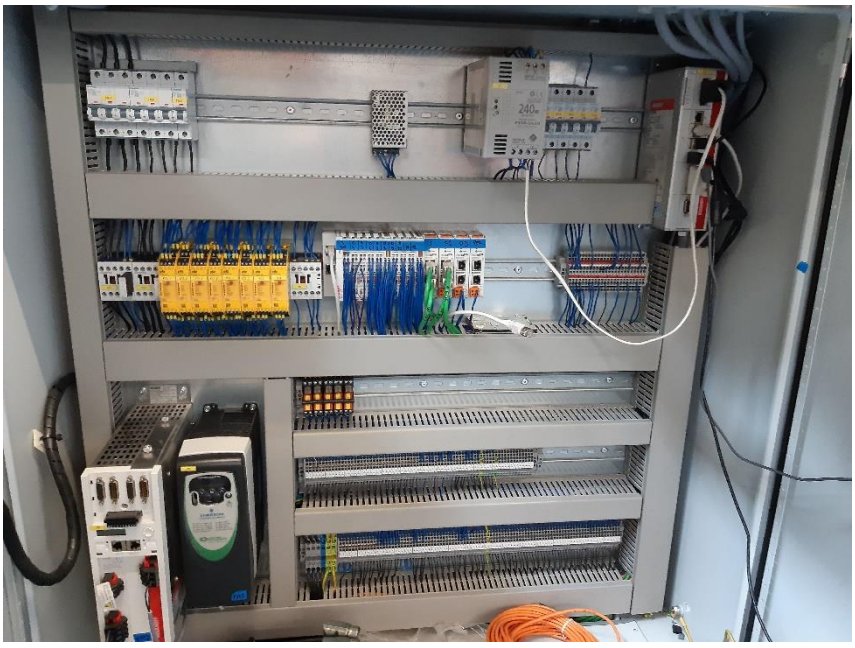
5.3 Řídící systém KUKA

Řídícím systémem robotu je KUKA KR C4. Jedná se o systém z roku 2013, který je však stále v prodeji. Kontroler má ochranu IP54 a pracuje při teplotách +5 až +45 °C. Rozměry kontroleru jsou 960 x 792 x 558 mm a váží 150 kg. Ke kontroleru je dodán nový pendant od firmy KUKA, zvaný smartPAD. [22] Kontroler je na obr. 20.

OBR. 20 KUKA KR C4

5.4 PLC Beckhoff

Jako PLC je na pracovišti použit kontrolní cabinet C6920-0040 od firmy Beckhoff z roku 2014. Základem PLC je CPU Intel Core i7 se čtyřmi jádry a frekvencí 2,3 GHz. Dále modul obsahuje dvě 2GB paměti DDR3L a 320 GB hard disk. Napětí na zdroji je 24 V. Programy poté běží v prostředí TwinCat 3. Popsané PLC je vidět v rozvaděči vpravo nahoře na obr. 21.



OBR. 21 ROZVADĚČ S PLC

5.5 Kamera a snímače

Snímání předmětů provádí kamera VCXG-125C.R od firmy Baumer s objektivem ZVL V1624-MPZ. Pro správné osvětlení snímané plochy slouží osvětlení DL-210 W od výrobce Smartview. Osvětlení produkuje bílé světlo a má aktivní plochu o průměru 210 mm.

Snímače byly vybrány od firmy SICK a jedná se o optoelektronické snímače řady W11 – 2 s odrazkou. Jde o reflexní světelnou závoru se standardní optikou. Rozměry snímačů jsou 15,6 mm x 48,5 mm x 42 mm. Snímací vzdálenost se pohybuje v rozmezí od 0,15 m po 10 m. Snímač využívá viditelné červené světlo o vlnové délce 640 nm, které vysílá LED dioda. Použitý snímač je na obr. 22. [23]

OBR. 22 SNÍMAČ SICK W11

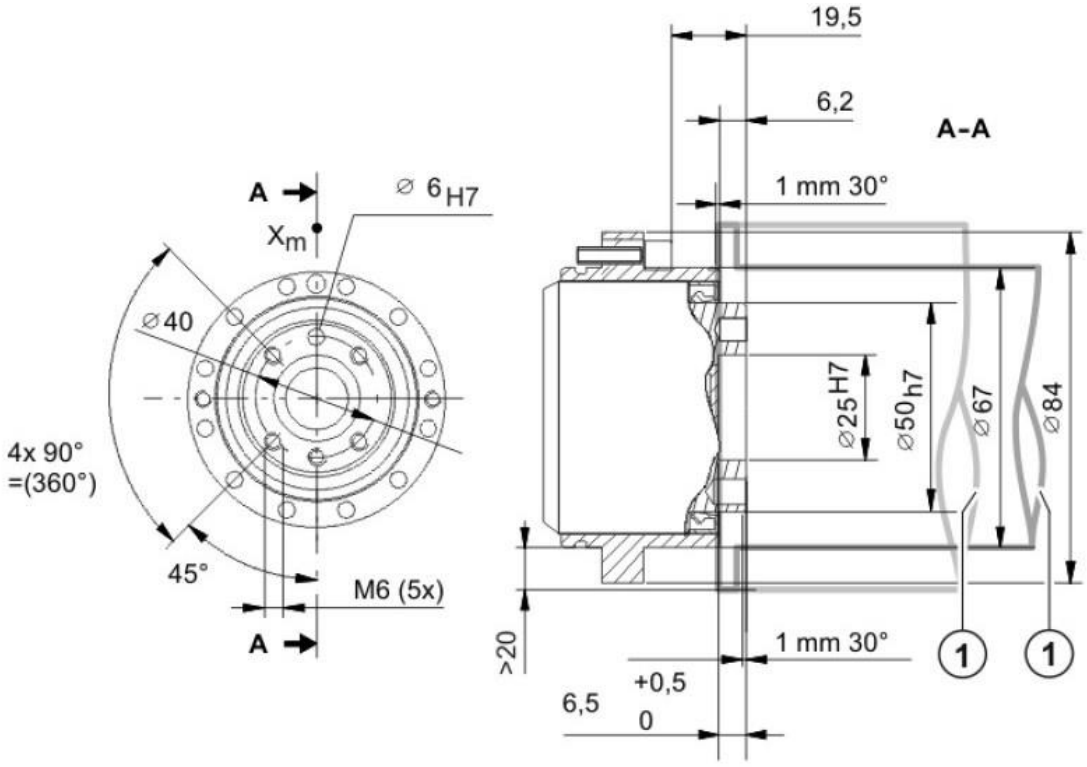
6 NÁVRH KONCOVÉHO EFEKTORU

Koncový efektor robotu je složen z příruby, přísavky a kalibračního kusu. Příruba je pak tvořena ze dvou dílů, a to z důvodu použití kalibračního kusu, který slouží ke kalibraci souřadných systémů robotu a kamery. Tento kus se nedá připevnit přímo na robot, proto musí být na přírubě. Zároveň však byl požadavek na vytvoření univerzální příruby, aby se v budoucnu dal efektor změnit z přísavky například na gripper. Z těchto důvodů je příruba dělená na dvě části. Jedna část příruby je tedy montována na robot a na druhou část příruby je pak montována přísavka. Koncový efektor, stejně jako všechny ostatní díly vytvořené pro účely této práce, byl vymodelován v programu Solidworks.

6.1 Příruba k robotu

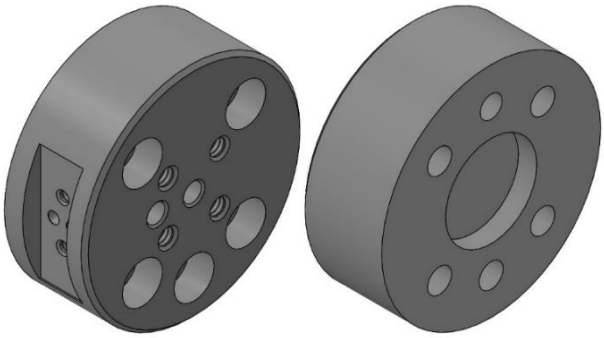
Příruba k robotu je klíčovou částí návrhu koncového efektoru. Při jejím návrhu je nutno vzít v potaz vícero faktorů. Základní tvar a rozměry určuje samotná příruba robotu, jejíž výkres od výrobce robotu je na obr. 23. Její tvar, montážní závity a díry pro středící kolíky popisuje obr. 12 (rozměry na obrázku jsou v mm). Příruba k robotu tedy musí obsahovat díru pro středící kolíky ve středu a v horní části příruby. Dále pět děr pro šrouby velikosti M6.

Průměr první části příruby k robotu je 60 mm. Na čelní straně příruby se pak nachází pět děr pro šrouby s válcovým zahloubením, jelikož pro upevnění příruby k robotu je použito pět imbusových šroubů velikosti M6. Dále obsahuje po vzoru montážního výkresu robotu díru pro kolík velikosti 25 mm ve středu a druhou díru pro kolík velikosti 6 mm. Díry pro kolíky mají tolerance H7 a nachází se na zadní straně příruby. Poloha všech děr odpovídá výkresu robotického zakončení.



OBR. 23 MONTÁŽNÍ VÝKRES PŘÍRUBY NA ROBOTU

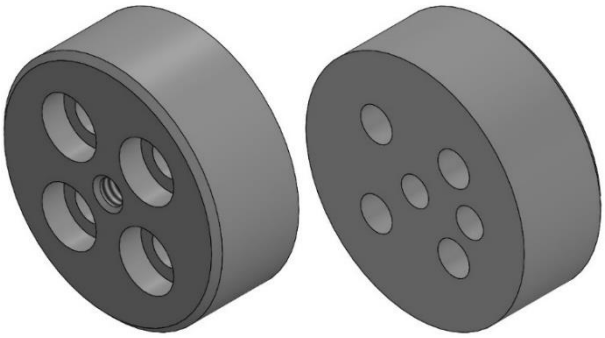
Na přední straně první části příruby se pak nachází dvě díry pro středící kolíky o velikosti 5 mm s tolerancí H7 a čtyři závitové díry o velikosti závitu M5. Všechny tyto díry slouží pro upevnění druhé části příruby, na kterou se montuje přísavka. Model první části příruby je ukázán na obr. 24.



OBR. 24 MODEL PRVNÍ ČÁSTI PŘÍRUBY

6.2 Příruba přísavky

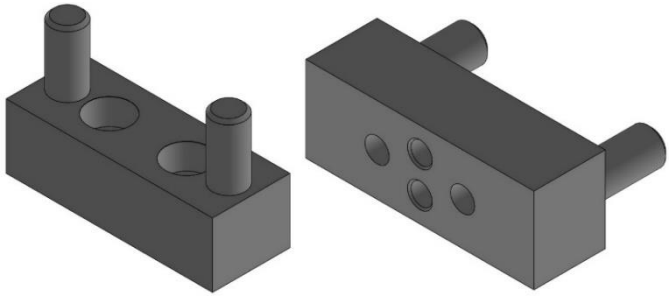
Druhá část příruby má na zadní straně díry pro středící kolíky stejné velikost a umístění jako první část. Na čelní straně se pak nachází čtyři díry pro šrouby s válcovým zahloubením pro šrouby o velikosti závitu M5. Ve středu čela je pak závitová díry o velikosti závitu M6 pro přichycení přísavky. Model druhé částí příruby je ukázán na obr. 25.



OBR. 25 MODEL DRUHÉ ČÁSTI PŘÍRUBY

6.3 Kalibrační kus

Kalibrační kus ne je umístěn na boku prvního dílu příruby. Díl slouží ke kalibraci souřadných systémů robotu a kamery, proto je u něj kladen důraz na přesnost. K přírubě je připevněn pomocí dvou šroubů M3 a vystředěn pomocí dvou kolíků o průměru 3 mm. Díl se bude zasouvat do protikusu, přimontovaného na dopravníku. Z tohoto důvodu jsou v něm zakomponovány dva trny o průměru 5 mm a délce 10 mm. Model kalibračního kusu je ukázán na obr. 26.



OBR. 26 KALIBRAČNÍ KUS

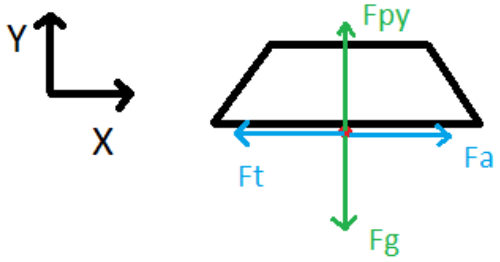
6.4 Vakuová přísavka

Vakuová přísavka byla vybrána z nabídky přísavek, které byly dostupné na dílně. Foto vybrané přísavky je na obr. 12. Jedná se o klasickou kruhovou přísavku o průměru 30 mm od firmy Festo. Výška gumové části přísavky je 10 mm, celková výška pak 40 mm. Napojení přívodní hadice je řešeno z boku a umožňuje napojit hadici o průměru 6 mm. Hmotnost přísavky je 36 g. Přísavka je vhodná především pro rovné povrchy, nicméně při zkoušce neměla problém s žádným s předmětů. Přísavka je vyfocena na obr. 27.

OBR. 27 VAKUOVÁ PŘÍSAVKA

6.4.1 Výpočet únosnosti

Pro zjištění vhodnosti přísavky je potřeba spočítat nejvyšší potřebnou sací sílu F_p pro nejtěžší předmět, kterým je největší kostka. Ta má hmotnost 0,362 kg. Pro výpočet sací síly je nejdříve spočítat její x a y složky. Výpočet zjednodušuje fakt, že všechny operace jsou prováděny s přísavkou orientovanou kolmo k zemi. Osa y je uvažována ve směru osy přísavky podle obr. 28.



OBR. 28 SCHÉMA SIL

Výpočet třecí síly F_t ve směru osy x:

$$\Sigma Fx = 0 \qquad \textit{ROV. 1}$$

$$\overrightarrow{Fa} + \overrightarrow{Ft} = 0 \qquad \textit{ROV. 2}$$

$$Fa - Ft = 0 \qquad \textit{ROV. 3}$$

$$Ft = m \cdot a = 0{,}362 \cdot 5 = 1{,}81\ N \qquad \textit{ROV. 4}$$

Kde: F_a…reakční síla od zrychlení

F_t…třecí síla

m…hmotnost kostky

a…zrychlení

Výpočet x složky sací síly F_P:

$$Ft = Fpx \cdot f \quad \text{ROV. 5}$$

$$Fpx = \frac{Ft}{f} = \frac{1{,}81}{0{,}5} = 3{,}62\ N \quad \text{ROV. 6}$$

F_{px}…x složka sací síly

f…součinitel tření

Výpočet y složky sací síly F_P:

$$\Sigma F_y = 0 \quad \text{ROV. 7}$$

$$\overrightarrow{F_{py}} + \overrightarrow{F_g} = 0 \quad \text{ROV. 8}$$

$$F_{py} - F_g = 0 \quad \text{ROV. 9}$$

$$F_{py} = m \cdot g = 0{,}362 \cdot 9{,}81 = 3{,}55\ N \quad \text{ROV. 10}$$

F_{py}… y složka sací síly

F_g…gravitační síla

g…gravitační zrychlení

Výpočet celkové síly F_P:

$$F_{pcelk} = F_{px} + F_{py} = 3{,}62 + 3{,}55 = 7{,}17\ N \quad \text{ROV. 11}$$

$$F_p = F_{pcelk} \cdot k = 7{,}17 \cdot 2 = 14{,}34\ N \quad \text{ROV. 12}$$

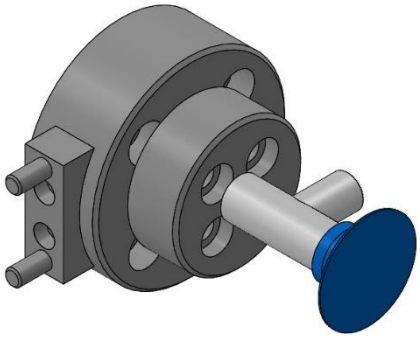
F_p…sací síla

k…koeficient bezpečnosti

Sací síla přísavky udávaná výrobcem je 40,8 N, což znamená, že přísavka je značně předimenzovaná. To však pro danou aplikaci není problém.

6.5 Finální efektor

Při montáži efektoru se nejdříve připevní za pomocí dvou šroubů M3 a dvou středících kolíků průměru 3 mm kalibrační kus k první části příruby. Poté je tato sestava namontována na robot. Vystředění zajistí kolíky 25 mm a 6 mm, připevnění pak pět šroubů M6 s imbusovou hlavou. Dalším krokem je montáž druhé části příruby na první část. Jako spojovací materiál jsou použity čtyři šrouby M5 a díly jsou vůči sobě dány do pozice díky středícím kolíkům velikosti 5 mm. Poté se přimontuje přísavka k druhému dílu příruby za pomoci stavěcího šroubu M5 bez hlavy. Ten se zašroubuje do druhého dílu příruby a na něj se poté přišroubuje samotná přísavka. Poté již zbývá pouze připojit hadici pro přívod vzduchu. Model hotového koncového efektoru bez spojovacího materiálu je na obr. 29. Výkresy sestavy koncového efektoru jsou k dispozici v příloze 1.



OBR. 29 MODEL KONCOVÉHO

7 OPC UA KOMUNIKACE PLC A PROCESS SIMULATE

Pro virtuální zprovoznění pracoviště je potřeba vytvořit komunikaci mezi simulačním softwarem Process Simulate a řídícím PLC. Software Process Simulate pro tuto potřebu umožňuje více způsobů propojení. Pokud je však PLC nebo jiný externí prvek jiné značky než Siemens, nejjednodušším a nejuniverzálnějším způsobem propojení jednotlivých programů je použití OPC komunikace.

OPC je otevřená komunikační platforma, jejímž účelem je propojení real time řídících systémů různých výrobců se SCADA a jinými HMI systémy. Hlavní využití této platformy je v průmyslových aplikacích, i když není omezena pouze na ně. Postupně do platformy byly implementovány technologie a platformy různých firem a jazyků, včele s .NET frameworkem od společnosti Microsoft, podporou XML a další. OPC platforma má více specifikací. Mezi nejznámější patří OPC DA. Tato specifikace zvládala komunikaci s většinou řídících systémů, nicméně byla převážně zaměřena na operační systém Windows od společnosti Microsoft. Z tohoto důvodu vznikla specifikace OPC UA, kde UA je anglická zkratka pro jednotnou architekturu, tudíž tento způsob komunikace může běžet i na ostatních operačních systémech jako Linux, Android nebo iOS. [24]

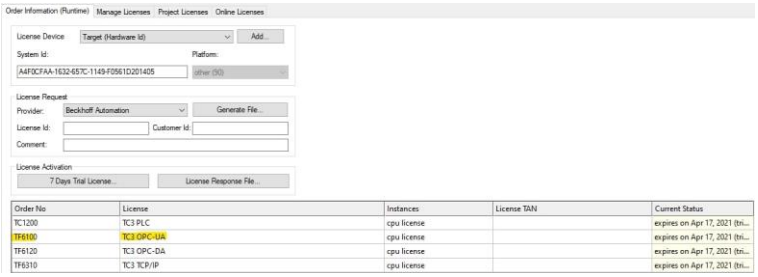
OPC UA využívá dvou základních protokolů. Jedním je binární protokol pro real time komunikaci je opc.tcp://Server, druhým pak webový protokol http://Server. Pro propojení PLC se simulačním softwarem je používán první typ protokolu.

7.1 Nastavení OPC serveru v PLC

Obecně OPC komunikace funguje na principu server/client propojení. Je tedy nutné vytvořit server na jedné straně a klienta na druhé. Server bývá většinou vytvořen v PLC, jako v případě této práce, případně externě pomocí k tomu určené aplikace. Pro umožnění OPC UA komunikace v prostředí TwinCAT 3 je nutné stáhnout ze stránek firmy Beckhoff doplněk TF6100-OPC-UA.4.3.26.0 (případně jinou verzi).

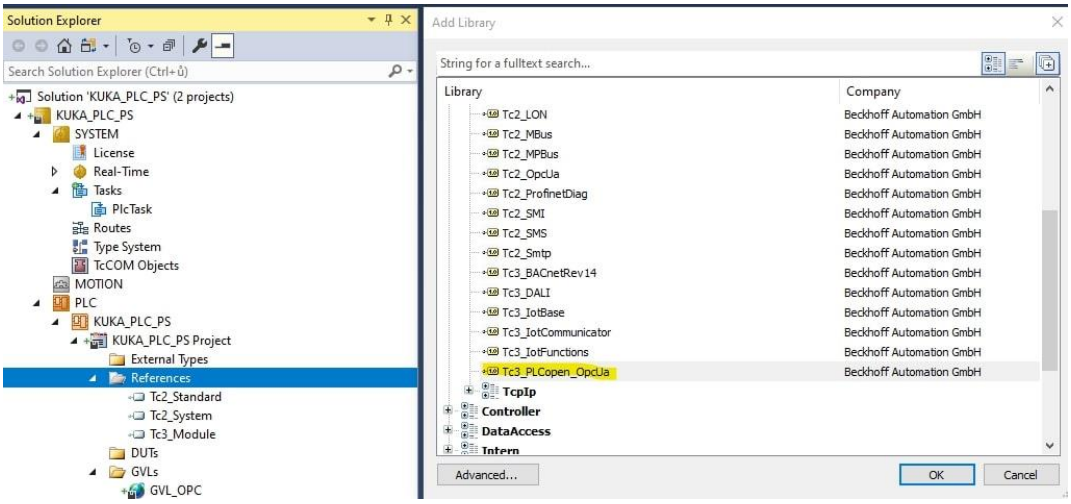
7.1.1 Přidání licencí a knihoven

Po založení nového projektu a PLC v prostředí TwinCAT 3 je potřeba projektu přiřadit licenci pro použití OPC UA komunikace. To je provedeno v okně licencí, které se otevře dvojklikem na ikonu *License* umístěnou ve stromu v Solution Exploreru pod ikonou *SYSTEM*. V záložce *Manage Licenses* vybereme licenci číslo TF6100 s názvem TC3 OPCC-UA. Přiřazení licence zkontrolujeme v záložce *Order Information* (Runtime) podle obr. 30. V případě potřeby vybereme i další licence. Pro testovací účely je použita sedmidenní trial licence.



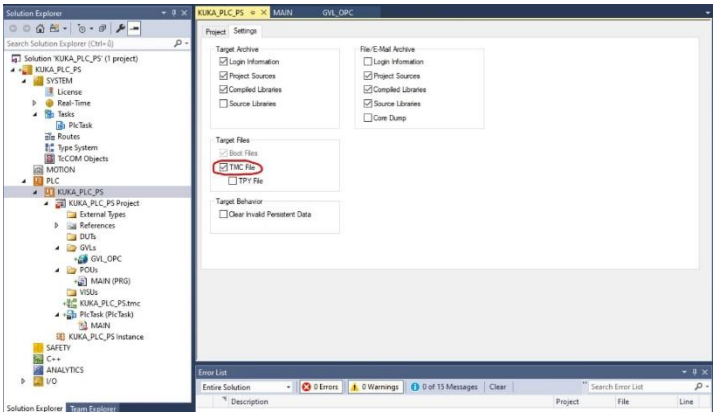
OBR. 30 LICENCE TC3 OPC-UA

Pro případ potřeby použití některých příkazů při pozdějším programování je také dobré importovat knihovnu Tc3_PLCopen_OpcUa ukázanou na obr. 31. Knihovny se přidávají kliknutím pravým tlačítkem myši na záložku *References* a vybráním *Add*. Poté se vybere potřebná knihovna.

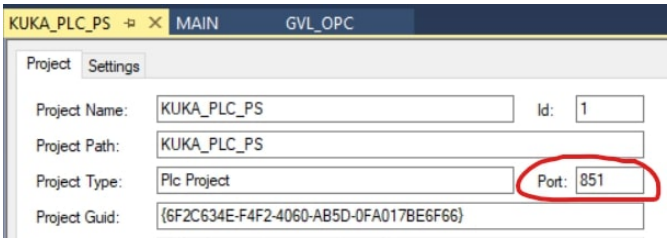


OBR. 31 IMPORT KNIHOVNY

Dalším důležitým krokem před samotnou tvorbou serveru je povolení přístupu TwinCATu k TMC souborům. Jedná se o soubory používané pro přenos informací a zpřístupní se v záložce *Settings* po kliknutí na název PLC v *Solution Eploreru*, jak ukazuje obr. 32. [25] Poté je dobré zkontrolovat, případně změnit port PLC v záložce *Project* podle obr. 33 Do té se dostane stejným způsobem jako do záložky *Settings*. V případě použití jednoho PLC může být použit port nastavený systémem. Při použití více PLC je dobré porty změnit, aby každé PLC používalo jiný port.



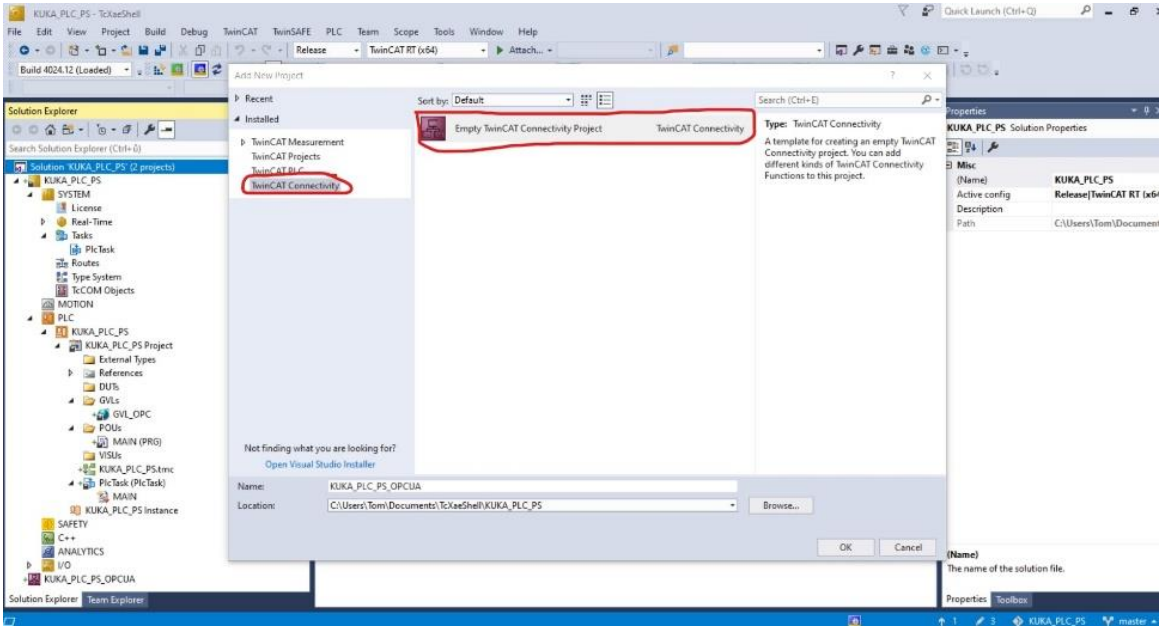
OBR. 32 ZPŘÍSTUPNĚNÍ TMC SOUBORŮ



OBR. 33 NASTAVENÍ PORTU POUŽÍVANÉHO PLC

7.1.2 Tvorba serveru

Dalším krokem je nastavení samotného serveru. Dá se vytvořit pro pomocí doplňku, který je součástí balíku TF6100-OPC-UA nebo přímo v softwaru TwinCAT 3. První možnost je jednodušší a lehce rychlejší. Druhá možnost však nabízí větší možnosti nastavení serveru, přehled a především možnost založení více serverů. To je přínosné například při větších projektech, kdy řízení zajišťuje více PLC a každé komunikuje s jiným dalším zařízením. Tímto způsobem jsou v každé komunikaci přenášeny pouze potřebné informace. Samotná tvorba serveru je složena z několika kroků. Nejdříve je potřeba založit nový komunikační projekt. K tomu je potřeba kliknout levým tlačítkem myši na *Solution* nahoře v *Solution Exploreru* a vybrat možnost *Add New Project*. Poté se vybere možnost *TwinCAT Connectivity* a založí se projekt obdobně jako při zakládání PLC projektu. Je dobré projekt uložit do stejného adresáře, v jakém se nachází projekt PLC, většinou se jedná o již přednastavené uložiště. Postup je ukázán na obr. 34.

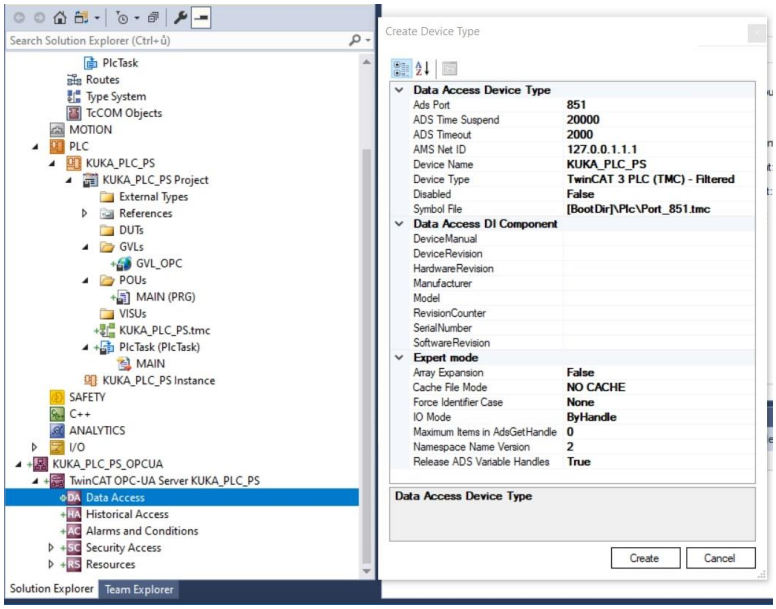


OBR. 34 TVORBA CONNECTIVITY PROJEKTU

Dále se přidá nový server. Tuto akci je potřeba provést manuálně právě kvůli možnosti tvorby více serverů. Přidání se provede kliknutím levým tlačítkem myši na nově vytvořený komunikační projekt, výběrem možnosti *Add New* a poté výběrem typu serveru. Pokud nejsou nainstalované jiné komunikační doplňky než TF6110-OPC-UA, měl by být výběr omezen pouze na *TwinCAT OPC-UA Server Project*.

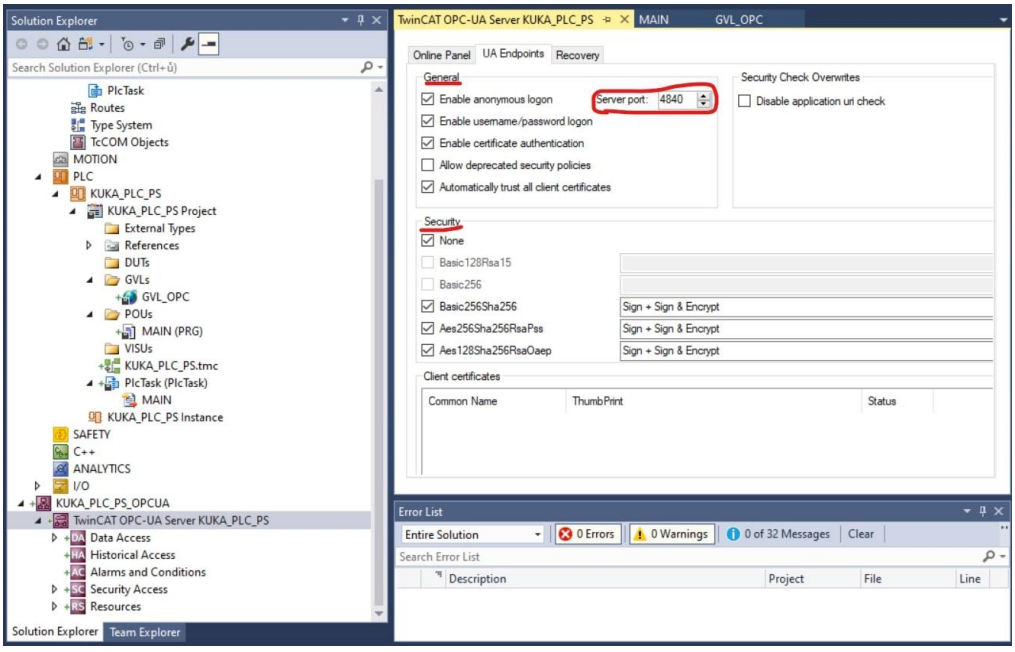
7.1.3 Definice serveru

Po vytvoření serveru je potřeba vytvořit konkrétní zařízení. To zastupuje jednotlivá PLC. Zároveň podle potřeby lze zařízení vytvořit pro konkrétní účel. Většinou bývá nejvhodnější možnost *Data Access*. Pravým tlačítkem myši tedy klikneme na tuto možnost a přidáme nové zařízení. V kartě, která se otevře, vyplníme název zařízení, jeho IP adresu a další potřebné údaje obdobně jako na obr. 35. Pokud je potřeba některé údaje změnit, stačí kliknout na dané zařízení a v levé části okna se zobrazí záložka *Properties*, kde lze údaje libovolně měnit.



OBR. 35 TVORBA ZAŘÍZENÍ

Vytvořené zařízením se dvojklikem otevře k úpravě a v záložce *UA Endpoints* se nastaví komunikační port serveru. V případě jednoho serveru může být ponechá přednastavený port. Dále se v této záložce nastavení zabezpečení serveru. Pro testovací účely lze vybrat možnost žádného zabezpečí, pro provoz však tato možnost není doporučovaná. Nastavení popisuje obr. 36.

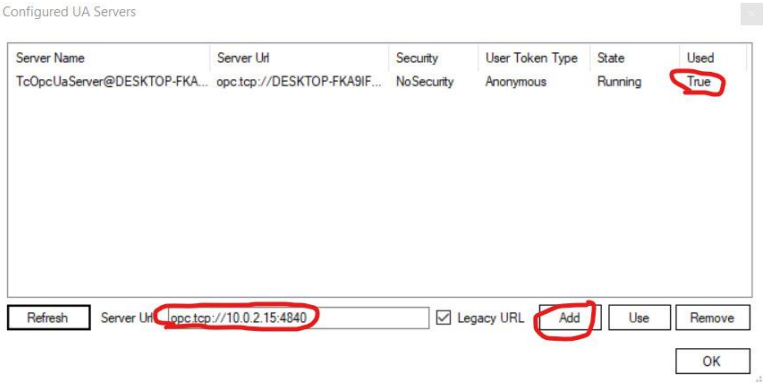


OBR. 36 NASTAVENÍ PŘÍSTUPU K SERVERU

Pro další práci se serverem a připojení k serveru je vhodné zobrazit *OPC UA Configurator* v horní liště záložek. Pro zobrazení záložky je potřeba najít v menu *View* možnost *Toolbars* a najít příslušnou záložku v nabídce, která se otevře. *OPC UA Configurator* nám umožní jednoduché a přehledné nastavení a spuštění serveru v prostředí TwinCAT 3.

7.1.4 Přidání serveru

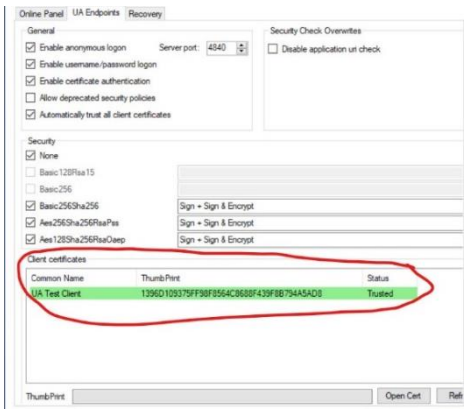
Nastavení spouštění serveru probíhá přes položku *Add Target*. Jedná se o první okno v záložce *OPC UA Configurator*. Po jeho rozkliknutí se objeví okno *Configured UA Servers*. Zde jsou zobrazeny všechny servery, ke kterým se může PLC připojit. Pro přidání nového serveru je potřeba vyplnit adresu serveru v liště *Server URL* a kliknout na tlačítka *Add*. Adresa serveru by měla mít podobu opc.tcp://IP:port. Postup je ukázán na obr. 37. Je důležité zkontrolovat, zda je status ve sloupci *Used* na hodnotě *True*. Pokud ne, lze status změnit kliknutím na tlačítko *Use*, umístěné vedle tlačítka *Add*. Vše se potvrdí tlačítkem *OK*.



OBR. 37 NASTAVENÍ SPOUŠTĚNÍ SERVERU

Po přidání serveru pak stačí v poli, kde bylo původně pouze *Add Target* vybrat nově přidaný server a ve vedlejším poli vybrat úroveň zabezpečení, kterou chceme použít a lze se připojit k serveru pomocí tlačítka *Connect*. Vyskočí autentizační okno, ve kterém je možnost přihlášení se jako uživatel nebo v případě testování a nastavení žádného zabezpečení se lze přihlásit jako anonymní uživatel. Po vybrání a potvrzení připojení může vyskočit okno *Load from target* s dotazem, jestli se má PLC připojit k exitujícímu připojení. Je lepší kliknout na *Ne* a spustí se nové připojení, které nebude závislé na předchozích připojeních.

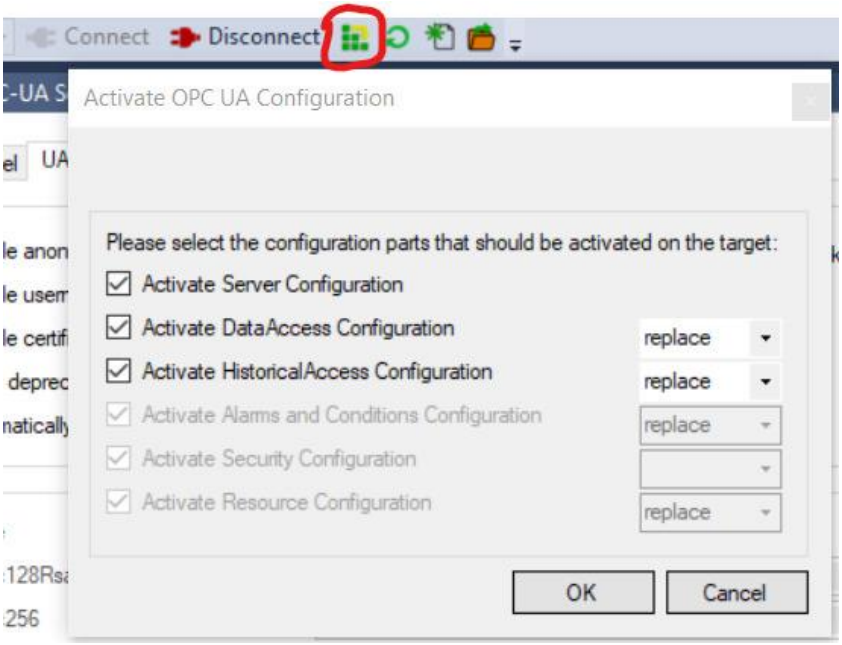
Po připojení k serveru je ještě potřeba otevřít nastavení serveru v komunikačním projektu a v záložce *UA Endpoints* zkontrolovat, jestli je v oblasti *Client certificates* přítomné naše spojení a zda je zeleně podbarveno, jak ukazuje obr. 38. Pokud je podbarveno červeně, mělo by stačit kliknutí na tlačítko *Refresh* k obnovení statusu.



OBR. 38 KONTROLA DŮVĚRYHODNOSTI

7.1.5 Aktivace serveru a definice proměnných

Na závěr je potřeba aktivovat OPC UA konfiguraci. Jedná se o obdobu aktivace konfigurace běžného PLC projektu. Aktivace proběhne kliknutí na zelenou ikonu vedle tlačítka pro odpojení serveru. Po rozkliknutí se ukáže okno *Activate OPC UA Configuration*. V něm je dobré zatrhnout všechny položky a z nabídek vybrat možnosti *replace*. Aktivace konfigurace je ukázána na obr. 39. Aktivaci konfigurace serveru je dobré dělat před každým spuštění a je potřeba ji udělat po každé proveden změně na serveru, aby byly všechny provedené změny provedeny po spuštění.



OBR. 39 AKTIVACE KONFIGURACE SERVERU

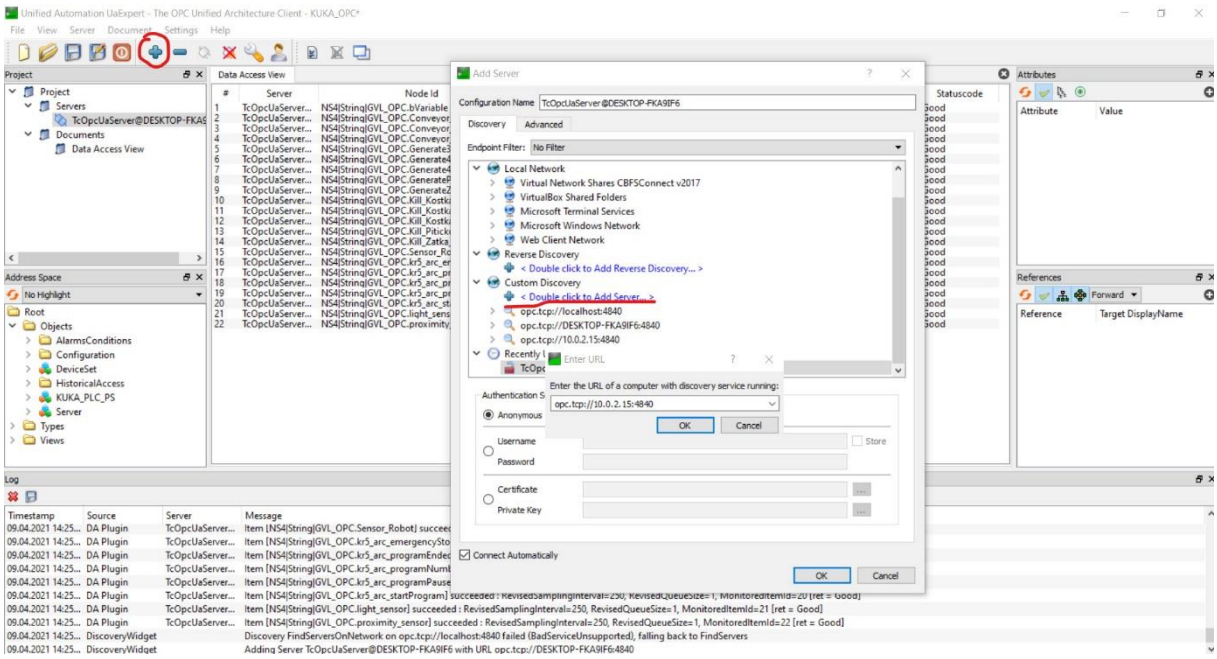
Když je server připojený, je ještě potřeba u proměnných v GVL přidat před každou proměnnou, kterou je potřeba zařadit do OPC komunikace vložit krátký identifikátor {attribute ´OPC.UA.DA´ := ´1´}. OPC UA komunikace umožňuje propojení jednotlivých signálů pomocí jejich názvů. Odpadá tak nutnost adresace jednotlivých proměnných. Příklad definice proměnných pro OPC komunikaci ukazuje obr. 40.

OBR. 40 DEFINICE PROMĚNNÝCH

7.2 Výběr a nastavení OPC klienta a propojení s PLC

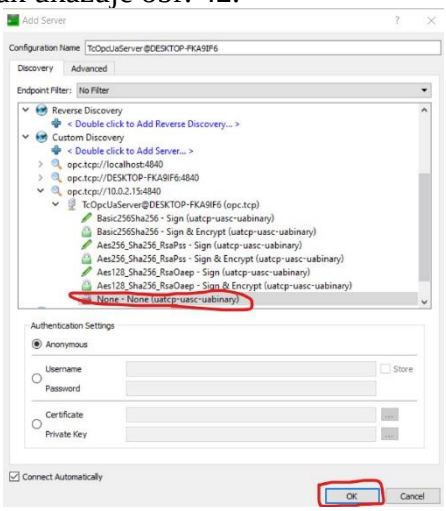
OPC klient slouží k připojení zařízení k serveru. Klient se dá nastavit více způsoby, nejjednodušší a nejčastějším však bývá využití programů třetích stran. Jelikož Beckhoff využívá spolupráce s Unified Automation byl zvolen klient této společnosti. Po stažení klienta jej stačí pouze zapnout, vyhledat server a připojit se k němu.

Přidání serveru se provádí kliknutím na tlačítko + v horní nabídce. Poté vyskočí okno Add Server. Zde se v sekci *Custom Discovery* dvakrát klikne na položku *Double click to Add Sever*. Vyskočí okno *Enter URL*. Lze zvolit, jaký typ OPC připojení chceme použít. Pro připojení k serveru PLC je potřeba zadat adresu v podobě opc.tcp://IP:port. Jedná se o stejnou adresu, která se zadávala v okně *Configured UA Servers* při přidávání serveru do PLC před jeho připojením. Postup je ukázán na obr. 41.



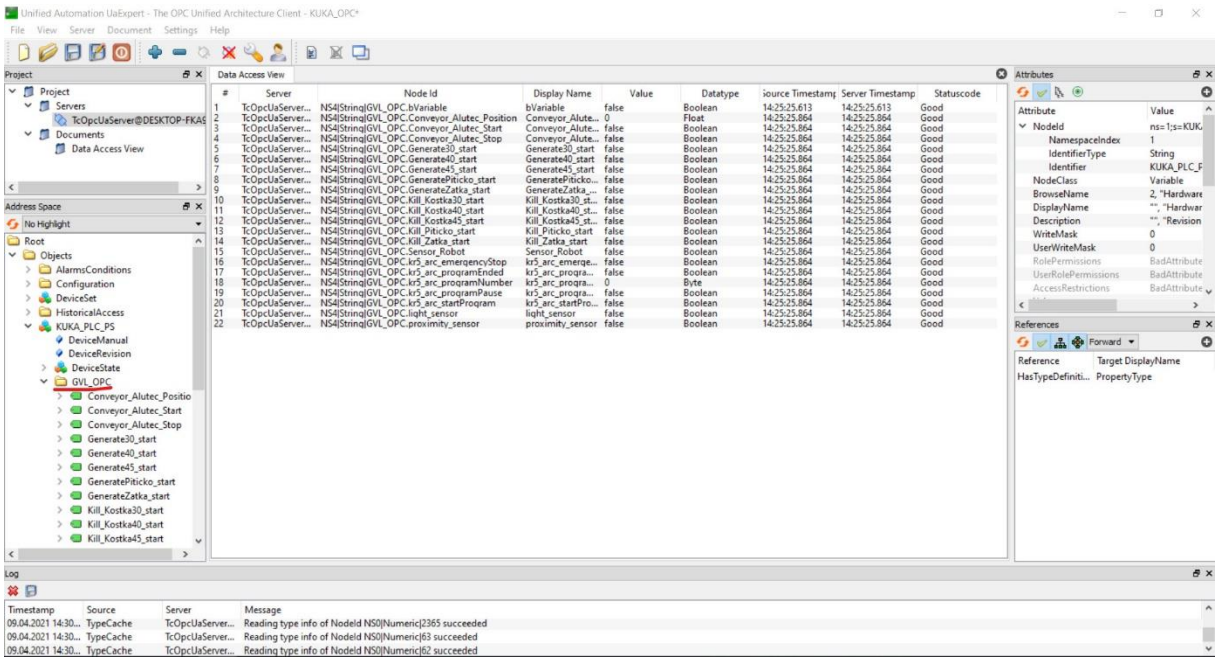
OBR. 41 PŘIDÁNÍ SERVERU

Po zadání adresy se server objeví v nabídce níže. Postupně je potřeba rozkliknout všechny nabídky nově přidaného serveru, dokud se neobjeví možnosti přihlášení. Pokud byla u serveru vytvořena možnost přihlášení bez zabezpečení pro testovací účely, lze ji nyní vybrat. A potvrdit připojení k serveru, jak ukazuje obr. 42.



OBR. 42 PŘIPOJENÍ K SERVERU

Připojení k server se dá vypínat nebo zapínat pomocí ikon v horní liště, umístěných vpravo od ikon přidání a odebrání serveru. Po připojení k serveru se v levé části okna v podokně Project objeví nově vytvořený server. Po jeho vybrání se o podokno níže zobrazí adresář serveru. Zde se postupně po otevření složek *Objects*, Název PLC serveru a Název GVL zobrazí proměnné z PLC. Jednoduchým přetažením do okna *Data Access View* je zobrazíme i s podrobnostmi a můžeme otestovat připojení, případně sledovat provoz. Zobrazené proměnné jsou na obr. 43.



OBR. 43 ZOBRAZENÍ PROMĚNNÝCH

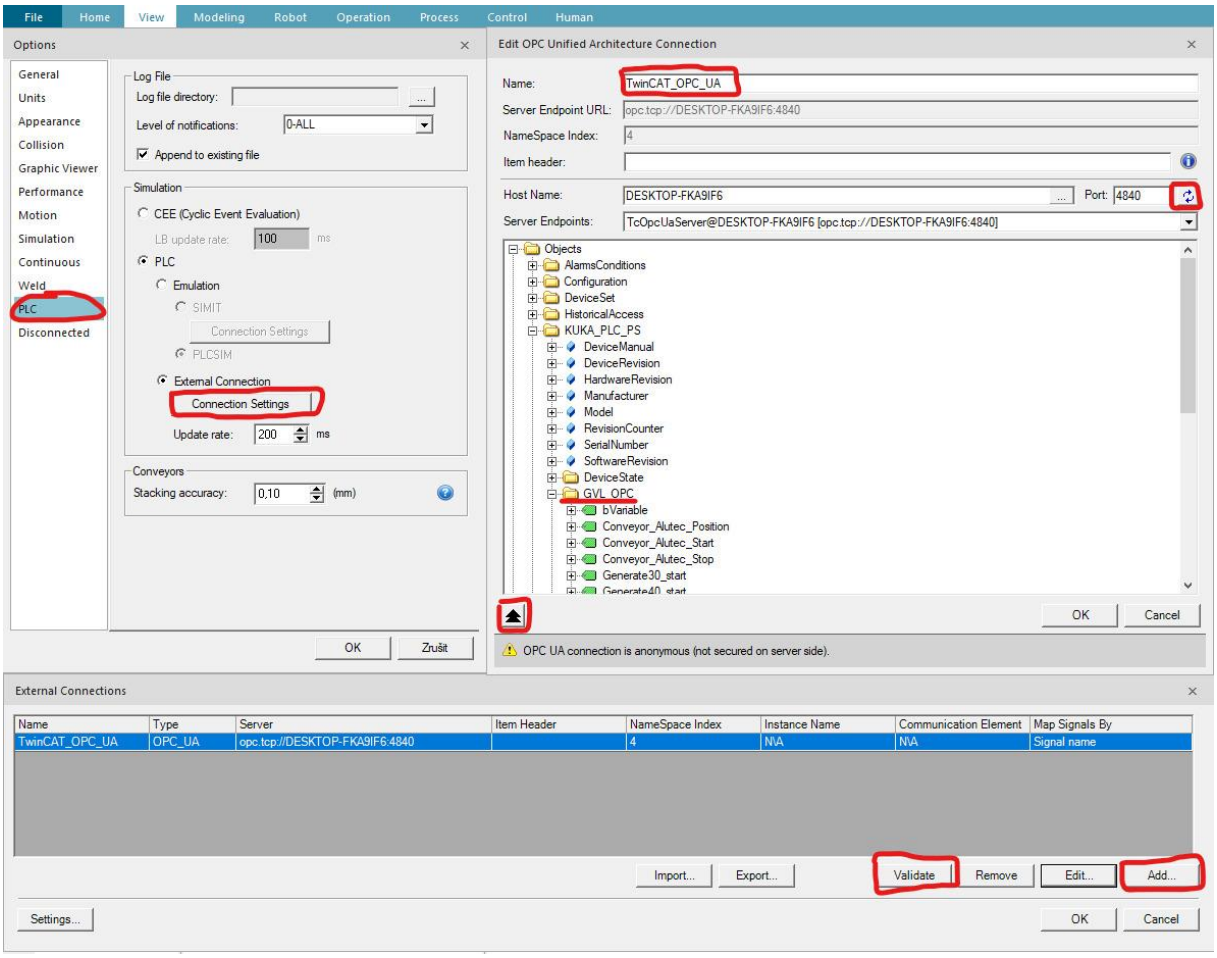
7.3 Propojení OPC serveru s Process Simulate

Připojení Process Simulatu k OPC UA serveru je v celku jednoduché. Pro správnou funkci je potřeba spustit Process Simulate jako správce. Pro samotné připojení k serveru je potřeba v hlavním menu rozbalit záložku *File* a v otevřeném okně Options se rozklikne záložka *PLC*. Zde je v sekci Simulation v základu nastavená možnost *CEE*. V tomto režimu Process Simulate nekomunikuje s žádnými dalšími programy a veškeré řízení probíhá uvnitř programu. Je tedy nutné tento způsob simulace přepnout na simulaci *PLC*. Z následujících možností *Emulation* a *External Connection* je potřeba zakliknout druhou zmíněnou možnost.

Poté se objeví okno *External Connections*. Zde se po kliknutí na tlačítko *Add* vybere možnost *OPC UA* a otevře se okno *Add OPC Unified Architecture Connection*. Zde je potřeba vyplnit kolonku *Server Endpoint URL*, do které se napíše adresa v podobě opc.tcp://IP:port, obdobně jako při připojení OPC klienta od Unified Automation. Poté stačí kliknout na černou dvoj šipku vlevo dole a Process Simulate by si měl automaticky najít veškeré potřebné informace. Pro zobrazení informací a proměnných z OPC Serveru je ještě potřeba načíst obsah OPC UA serveru pomocí modrých šipek umístěných vpravo od čísla portu.

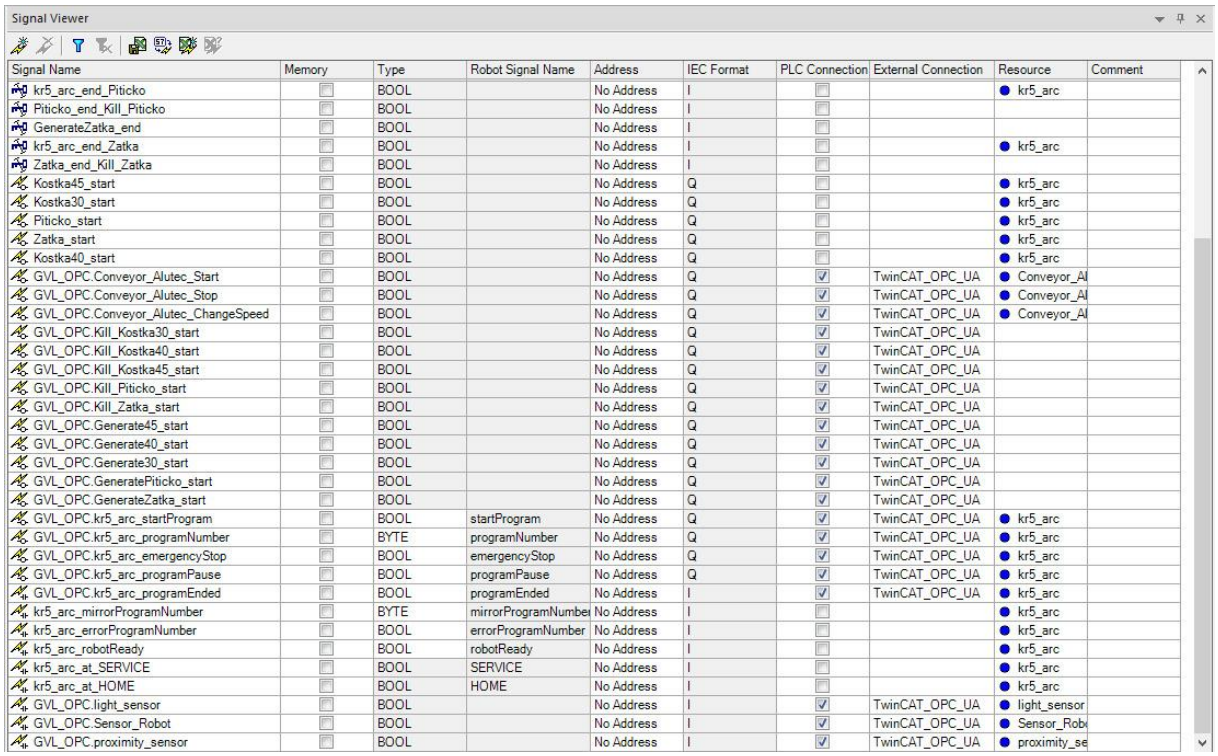
Následně lze v nově zobrazeném adresáři rozkliknout postupně složky *Objects*, Název PLC serveru a Název GVL, obdobně jako u OPC klienta od Unified Automation a zkontrolovat, zda jsou přítomny všechny proměnné.

Po přidání a připojení se k serveru je dobré vyzkoušet připojení k serveru pomocí tlačítka *Validate*. Celý postup dokumentuje obr. 44.



OBR. 44 PŘIPOJENÍ PROCESS SIMULATE K SERVERU

Po připojení k serveru je potřeba zvolit a upravit proměnné, které budou součástí OPC komunikace. Pro práci se signály slouží v Process Simulate okno *Signal Viewer*, které se zobrazí přes menu *View* a volbu Viewers. Připojení signálů ke komunikaci probíhá ve třech krocích. Nejdříve je potřeba u vybraných signálů zaškrtnou políčko ve sloupci PLC Connection. Poté se musí ve vedlejším sloupci External Connection vybrat komunikace, ke které chceme signál přiřadit. Lze totiž provozovat více komunikací. Na závěr je potřeba před názvy signálů, které chceme ke komunikaci přiřadit, přidat název konkrétního GVL, ve kterém se v PLC nachází. Zde lze signály exportovat do excelu a upravit v něm, což je vhodné při větším počtu signálů, nebo názvy změnit přímo v okně Signal Viewer. Po těchto úpravách jsou proměnné připraveny. Příklad proměnných přiřazených k OPC UA komunikaci je na obr. 45.


Signal Viewer

Signal Name	Memory	Type	Robot Signal Name	Address	IEC Format	PLC Connection	External Connection	Resource	Comment
kr5_arc_end_Piticko	☐	BOOL		No Address	I	☐		kr5_arc	
Piticko_end_Kill_Piticko	☐	BOOL		No Address	I	☐			
GenerateZatka_end	☐	BOOL		No Address	I	☐			
kr5_arc_end_Zatka	☐	BOOL		No Address	I	☐		kr5_arc	
Zatka_end_Kill_Zatka	☐	BOOL		No Address	I	☐			
Kostka45_start	☐	BOOL		No Address	Q	☐		kr5_arc	
Kostka30_start	☐	BOOL		No Address	Q	☐		kr5_arc	
Piticko_start	☐	BOOL		No Address	Q	☐		kr5_arc	
Zatka_start	☐	BOOL		No Address	Q	☐		kr5_arc	
Kostka40_start	☐	BOOL		No Address	Q	☐		kr5_arc	
GVL_OPC.Conveyor_Alutec_Start	☐	BOOL		No Address	Q	☑	TwinCAT_OPC_UA	Conveyor_Al	
GVL_OPC.Conveyor_Alutec_Stop	☐	BOOL		No Address	Q	☑	TwinCAT_OPC_UA	Conveyor_Al	
GVL_OPC.Conveyor_Alutec_ChangeSpeed	☐	BOOL		No Address	Q	☑	TwinCAT_OPC_UA	Conveyor_Al	
GVL_OPC.Kill_Kostka30_start	☐	BOOL		No Address	Q	☑	TwinCAT_OPC_UA		
GVL_OPC.Kill_Kostka40_start	☐	BOOL		No Address	Q	☑	TwinCAT_OPC_UA		
GVL_OPC.Kill_Kostka45_start	☐	BOOL		No Address	Q	☑	TwinCAT_OPC_UA		
GVL_OPC.Kill_Piticko_start	☐	BOOL		No Address	Q	☑	TwinCAT_OPC_UA		
GVL_OPC.Kill_Zatka_start	☐	BOOL		No Address	Q	☑	TwinCAT_OPC_UA		
GVL_OPC.Generate45_start	☐	BOOL		No Address	Q	☑	TwinCAT_OPC_UA		
GVL_OPC.Generate40_start	☐	BOOL		No Address	Q	☑	TwinCAT_OPC_UA		
GVL_OPC.Generate30_start	☐	BOOL		No Address	Q	☑	TwinCAT_OPC_UA		
GVL_OPC.GeneratePiticko_start	☐	BOOL		No Address	Q	☑	TwinCAT_OPC_UA		
GVL_OPC.GenerateZatka_start	☐	BOOL		No Address	Q	☑	TwinCAT_OPC_UA		
GVL_OPC.kr5_arc_startProgram	☐	BOOL	startProgram	No Address	Q	☑	TwinCAT_OPC_UA	kr5_arc	
GVL_OPC.kr5_arc_programNumber	☐	BYTE	programNumber	No Address	Q	☑	TwinCAT_OPC_UA	kr5_arc	
GVL_OPC.kr5_arc_emergencyStop	☐	BOOL	emergencyStop	No Address	Q	☑	TwinCAT_OPC_UA	kr5_arc	
GVL_OPC.kr5_arc_programPause	☐	BOOL	programPause	No Address	Q	☑	TwinCAT_OPC_UA	kr5_arc	
GVL_OPC.kr5_arc_programEnded	☐	BOOL	programEnded	No Address	I	☑	TwinCAT_OPC_UA	kr5_arc	
kr5_arc_mirrorProgramNumber	☐	BYTE	mirrorProgramNumber	No Address	I	☐		kr5_arc	
kr5_arc_errorProgramNumber	☐	BOOL	errorProgramNumber	No Address	I	☐		kr5_arc	
kr5_arc_robotReady	☐	BOOL	robotReady	No Address	I	☐		kr5_arc	
kr5_arc_at_SERVICE	☐	BOOL	SERVICE	No Address	I	☐		kr5_arc	
kr5_arc_at_HOME	☐	BOOL	HOME	No Address	I	☐		kr5_arc	
GVL_OPC.light_sensor	☐	BOOL		No Address	I	☑	TwinCAT_OPC_UA	light_sensor	
GVL_OPC.Sensor_Robot	☐	BOOL		No Address	I	☑	TwinCAT_OPC_UA	Sensor_Rob	
GVL_OPC.proximity_sensor	☐	BOOL		No Address	I	☑	TwinCAT_OPC_UA	proximity_se	

OBR. 45 POJMENOVÁNÍ A ÚPRAVA PROMĚNNÝCH

8 VIRTUÁLNÍ ZPROVOZNĚNÍ V PROCESS SIMULATE

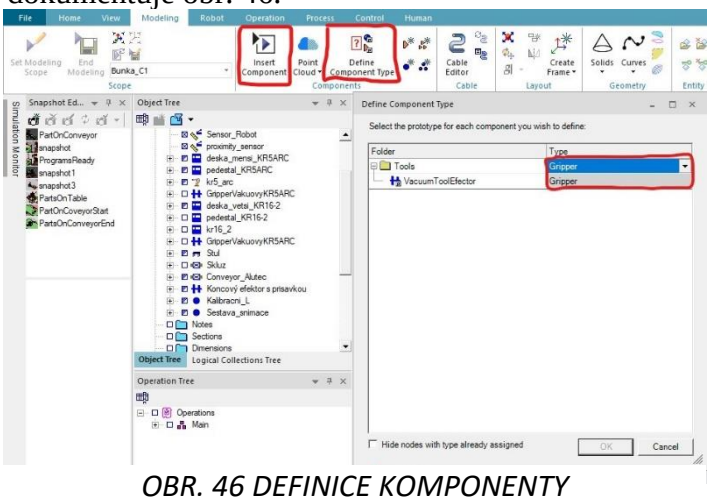
Původním plánem práce bylo vytvořit virtuální zprovoznění pracoviště, obsahujícího dopravník, robot a odkládací stůl. K řízení robotu měl být použit kontroler robotu, který by byl podřízený PLC. PLC by zpracovávalo data, která by dostalo z programu, který vyhodnocuje snímky kamery.

V průběhu práce se však ukázalo, že software Process Simulate není schopný měnit pozice generovaných dílů na základě souřadnic dodaných z PLC. Process Simulate umožňuje import souřadnic pomocí jednotlivých proměnných přes OPC UA komunikaci, dále však nejdou použít pro řízení jednotlivých bodů, tzv. Framů, které poté řídí generování materiálu. Z tohoto důvodu bylo nutné přistoupit k omezení simulace na robotickou část. Simulace je však stále přínosná, jelikož umožňuje rychlou kontrolu kolizí a především generování programů, které se po úpravě dají použít pro řízení skutečného robotu, bez nutnosti definice nových bodů.

8.1 Tvorba modelu a definice kontroleru

Základ pracoviště byl převzat z již exitujícího modelu, který bude brzy přístupný pro studijní účely studentům UVSSR. Bylo však potřeba provést úpravy layoutu. Prvním krokem tedy bylo odstranění nebo skrytí přebytečných prvků pracoviště, které se netýkají operací popsaných v této práci. Jedinou nepřesností ponechanou na modelu pracoviště bylo ponechání optických závor na odkládacím stolu, jelikož model stolu neumožňoval jejich skrytí či odstranění. Závory byly ponechány, jelikož nemají vliv na samotnou simulaci.

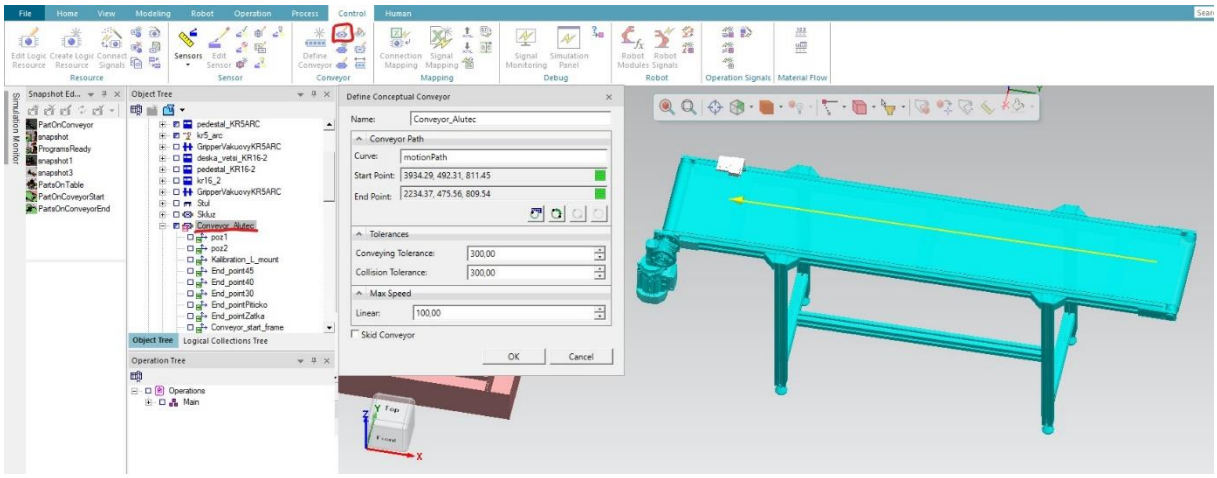
Druhým krokem bylo přidání potřebných komponent a vložení součástí pro manipulaci. Všechny součásti a komponenty se importují podobným způsob, proto je uveden pouze příklad importování koncového efektoru. Napřed je potřeba definovat typ vkládané komponenty. Pro možnost vložení komponenty je potřeba model komponenty vložit do složky s názvem NazevSoucasti.cojt. Tato složka musí být uložena v kořenovém adresáři Process Simulate, ideálně ve složce *Library*. Samotná definice se provádí v menu *Modeling*, kde se v sekci *Components* klikne na tlačítko *Define Component Type*. Poté, se v adresáři vybere umístění složky s modelem. Po výběru složky je otevřeno okno definice komponenty. Zde je zvolen typ komponenty, v tomto případě se jedná o *Gripper*. V případě součásti pro manipulaci se volí typ *PartPrototype*. Vše dokumentuje obr. 46.



OBR. 46 DEFINICE KOMPONENTY

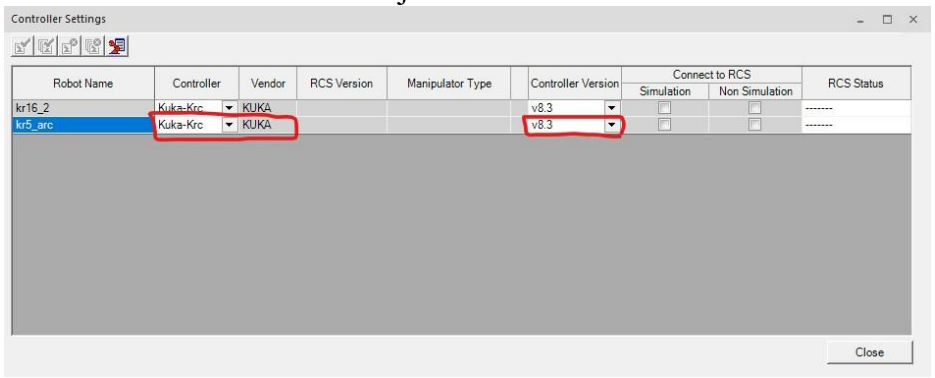
Když jsou všechny komponenty vloženy, následuje jejich umístění na pracoviště K tomuto účelu slouží příkazy *Placement Manipulator* a *Relocate*, které lze nalézt v nabídce po kliknutí pravým tlačítkem myši na součást buď v objektovém stromu nebo přímo v modelu. Výjimkou je koncový efektor robotu, který je k robotu přiřazen pomocí příkazu *Mount Tool*. Ten se nachází v nabídce po kliknutí pravým tlačítkem myši na robot.

Po rozmístění a přiřazení všech komponent a součástí je potřeba definovat jednotlivé komponenty. Nejdříve tedy byla provedena definice dopravníku. Pro definování dopravníku je nejdříve nutné jej uvést do rozmodelovaného stavu a vytvořit na něm dráhu, po které se má součást pohybovat. To se provádí příkazem *Set Modeling Scope* v menu *Modeling* po vybrání komponenty. Dráhu pak lze vytvořit pomocí příkazu *Polyline* v nabídce *Curves*, která se nachází v sekci *Geometry* v menu *Modeling*. Následně se při vybraném dopravníku ve stromu objektů pomocí příkazu *Define Conveyor*, který je umístěný v sekci *Conveyor* v menu *Control*, otevře okno *Define Conceptual Conveor*. Zde se zvolí v poli *Curve* dříve vytvořená dráha a mohou se nastavit parametry jako rychlost dopravníku, směr pohybu nebo šířka pohyblivé oblasti. Postup je ukázán na obr. 47.



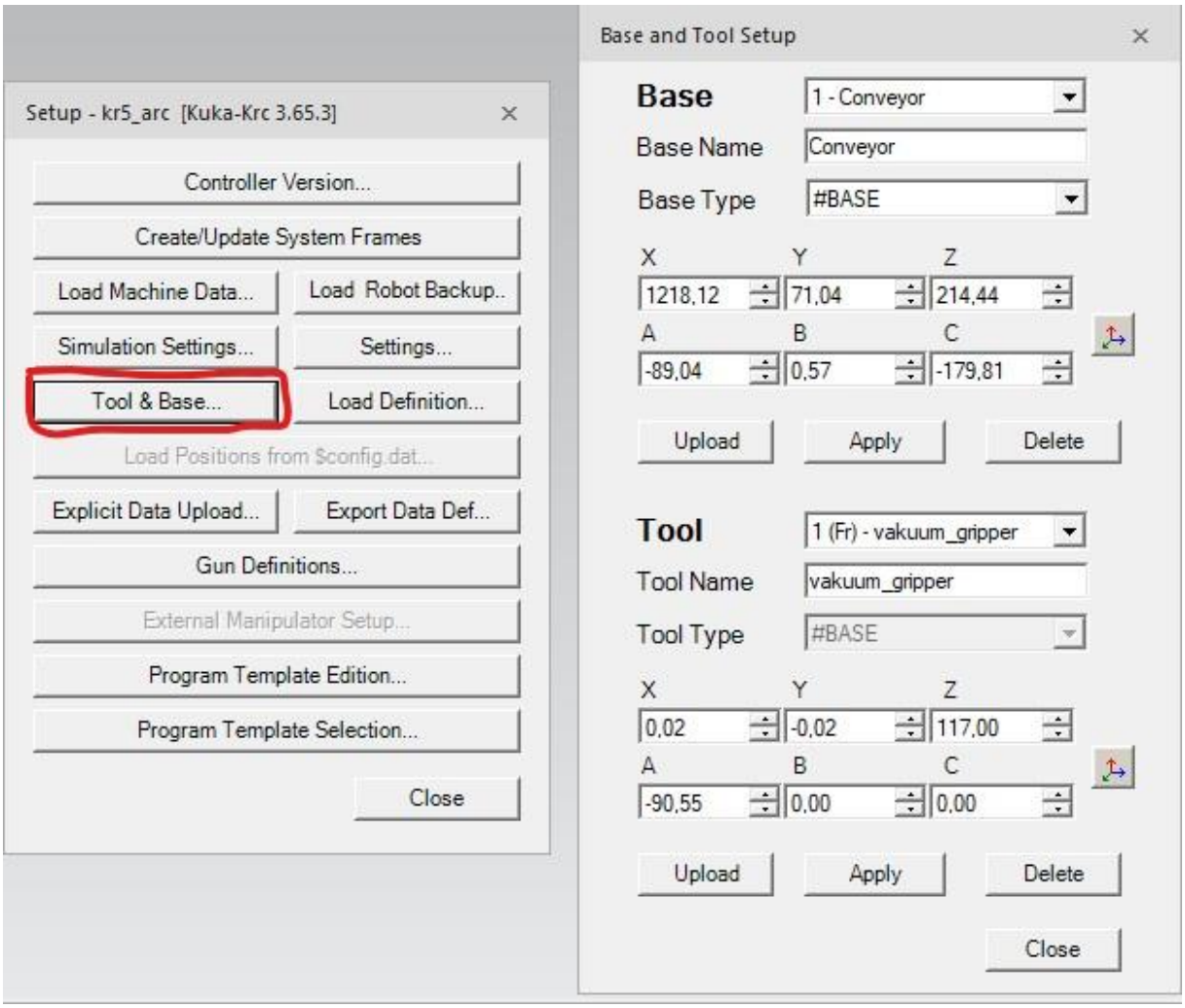
OBR. 47 DEFINICE DOPRAVNÍKU

Dále je potřeba zkontrolovat, případně definovat, kontroler, zda souhlasí s verzí kontroleru použitého na reálném pracovišti v dílně UVSSR. Okno nastavení kontroleru se otevře pomocí příkazu *Controller Settings* v sekci *Setup* v menu *Robot*. V okně Controller Settings jsou pak zobrazeny všechny komponenty, které jsou definovány jako robot. Pokud jsou v Process Simulate nainstalovány plug-iny obsahující kontrolery jednotlivých výrobců robotů, lze ve sloupci Controller vybrat kontroler příslušného výrobce. Dále ve sloupci *Controller Version* lze vybrat konkrétní verzi řídícího software, v případě kontroleru na pracovišti se jedná o verzi v8.3. Příklad nastavení kontroleru je na obr. 48.



OBR. 48 DEFINICE KONTROLERU ROBOTU

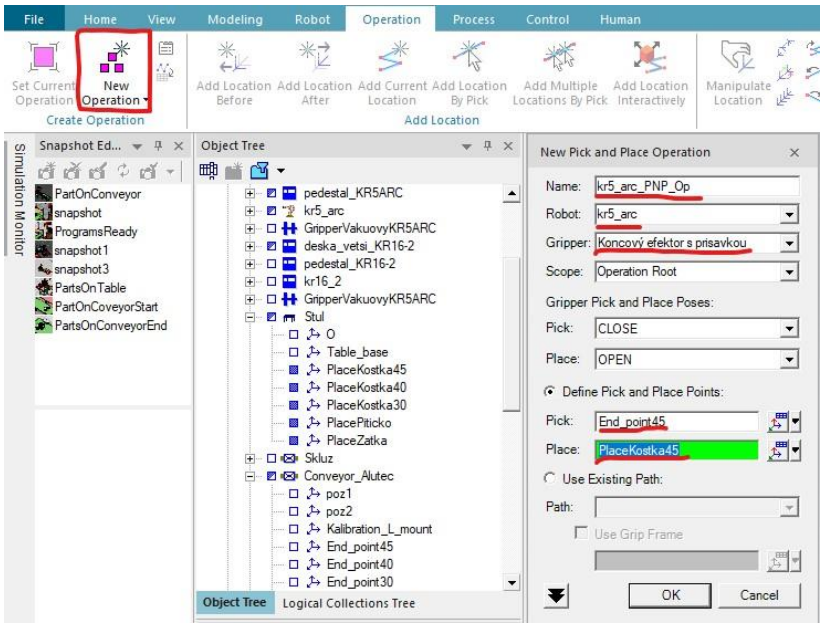
Na závěr přípravy pracoviště je dobré definovat nástroje a báze robotu. Jejich definice je obdobou definice těchto prvků pomocí teach pendantu reálného robotu a po přiřazení jednotlivých nástrojů a bází konkrétním bodům při vytváření drah a generování programu je možné zde vytvořené nástroje a báze převést a použít na skutečném robotu. Pro definici nástrojů a bází je nutné otevřít kartu nastavení robotu. To se dá provést více způsoby. Jedním z nich je vybrání robotu a poté v menu *Robot* a sekci *Setup* kliknout na *Robot Setup*. V okně *Setup* je pak potřeba vybrat možnost *Tool & Base*. Otevře se okno *Base and Tool Setup*, ve kterém lze nastavit jednotlivé bázové body, případně definovat nové nástroje. Postup ukazuje obr. 49.



OBR. 49 DEFINICE NÁSTROJŮ A BÁZÍ

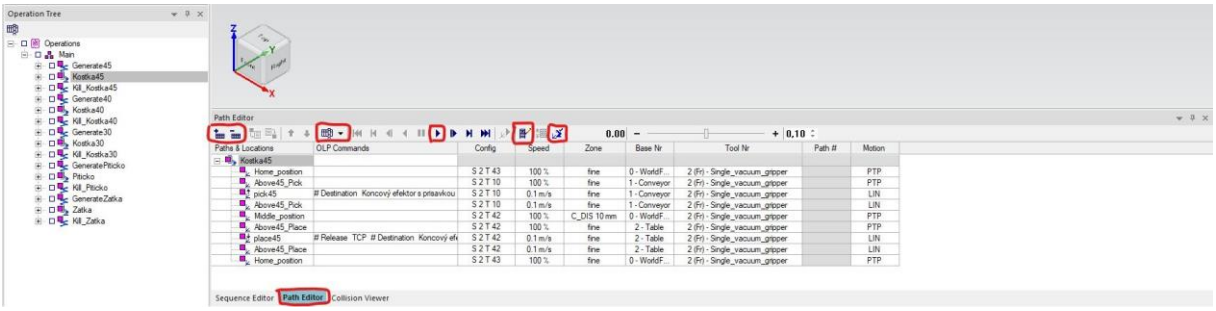
8.2 Tvorba a úprava operací

Když je layout pracoviště hotový a všechny kontrolery, nástroje a komponenty definovány, lze přistoupit k tvorbě jednotlivých pohybů robotu a ostatních částí. Simulace se zabývá jednoduchou Pick and Place operací provedenou pro více součástí. Nová Operace se založí v menu *Operation*, rozkliknutím možnosti *New Operation* a vybráním *New Pick and Place Operation*. Objeví se okno tvorby operace, v němž se operace pojmenuje, vybere se robot, který bude manipulaci provádět, případně se zvolí koncový efektor, pokud je robot vybaven více efektory. Dále se vyberou body vyzvednutí a položení součásti, případně se použije již připravená trajektorie. Postup ukazuje obr. 50.



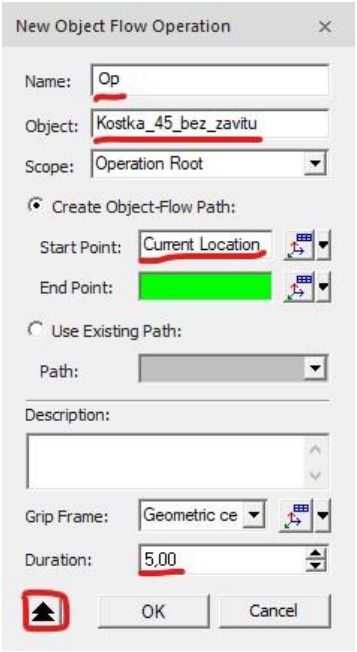
OBR. 50 TVORBA PICK AND PLACE OPERACE

Nově vytvořená operace se zobrazí v okně *Operation Tree*. Kliknutím na ikonu *Add Operations to Editor* v levém horním rohu okna *Path Editor* otevřeme operaci k editaci. Lze zde přidávat body do trajektorie pohybu, upravovat pořadí a pozici jednotlivých bodů a v neposlední řadě každému pohybu přiřadit atributy jako je rychlost, zóna, případně přiřazení konkrétního nástroje nebo báze. Jednotlivé sloupce pro možné úpravy se přidávají pomocí tlačítka *Customize Columns* v horní liště okna. Dále se zde dá spustit a ovládat simulace operace nebo nastavit parametry pro více bodů najednou, pomocí tlačítka *Set Locations Properties*. V neposlední řadě lze po vyzkoušení všech pohybů pomocí tlačítka *Auto Teach* a spuštěním simulace naučit robot konkrétní konfigurace jednotlivých os. Úpravu operace v Path Editoru ukazuje obr. 51.



OBR. 51 PŘÍPRAVA ROBOTICKÉHO PROGRAMU

Po vytvoření a úpravě všech pohybových operací je na řadě generování součástí. Postup tvorby operace je obdobný jako u Pick and Place operace, ovšem při výběru operace z nabídky *New Operation* se vybere možnost *New Object Flow Operation*. V nově otevřeném okně tvorby operace je možnost vyplnění názvu operace, výběru součásti, která bude generována výběr místa, kde bude součást generována a čas operace, který udává takt, jak často může být součást generována. Jako umístění součásti může být použita její současná poloha, podobně jako na obr. 52.

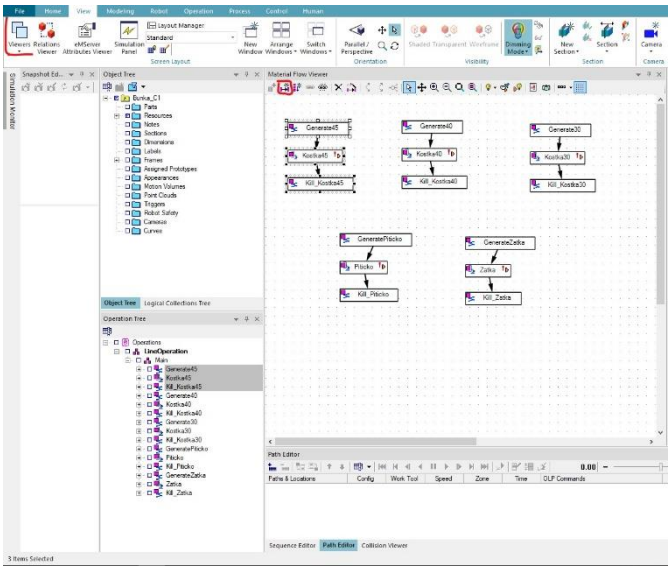


OBR. 52 GENEROVÁNÍ SOUČÁSTÍ

8.3 Materiálový tok a simulace

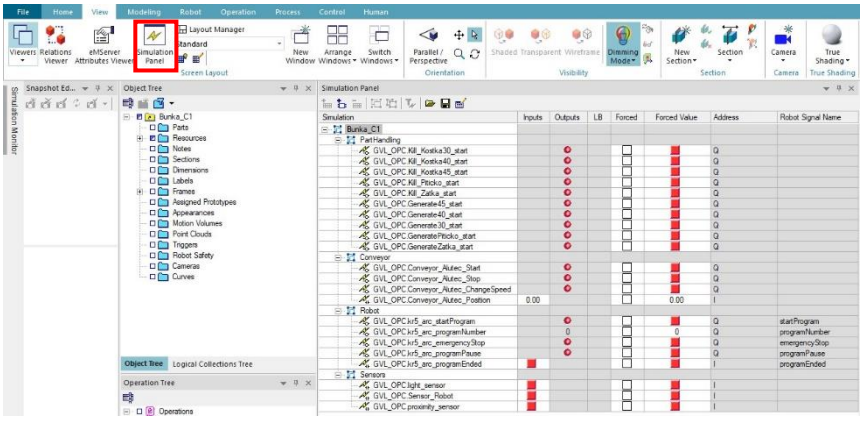
Před samotným spuštěním simulace je potřeba se přepnout do simulačního módu. To se provádí kliknutím na tlačítko *Line Simulation Mode*, umístěné v menu *Home* a sekci *Study*. Po přepnutí módu dojde k několika změnám, jako je například zmizení součástí pro manipulaci ze stromu objektů. Z tohoto důvodu se příprava pracoviště prováděla ve standardním módu.

Po přepnutí módu následuje tvorba materiálového toku. Ten je tvořen v okně *Material Flow Viewer*, které se zobrazí přes příkaz *Viewers* v menu *View*. Do okna materiálového toku přetáhneme z okna *Operation Tree* operace, u kterých je požadováno, aby na sebe logicky navazovaly. Po přetažení a uspořádání operací v okně je potřeba operace propojit pomocí tlačítka *New Material Flow Link*. Operace připravené pro simulaci ukazuje obr. 53.



OBR. 53 TVORBA MATERIÁLOVÉHO TOKU

Vytvořením materiálového toku je simulace připravená. Ke spuštění stačí všechny simulované operace Vložit do tzv. Compound operace v okně *Operation Tree* a tuto operaci přetáhnout do okna *Sequence Editor*, ve kterém se dají změnit délky operací, případně vytvořit spojení mezi operacemi, pokud simulace probíhá v *CEE* módu. Spuštění simulace pak probíhá v horní části okna. Průběh simulace lze sledovat v okně Simulation Panel, které se otevírá stejnojmenným tlačítkem v menu View a sekci Screen Layout. Jednotlivé signály se do okna importují přes ikonu Add Signal po vybrání signálu v okně Signal Viewer. Příklad použití je na obr. 54.



OBR. 54 SPUŠTĚNÍ SIMULACE

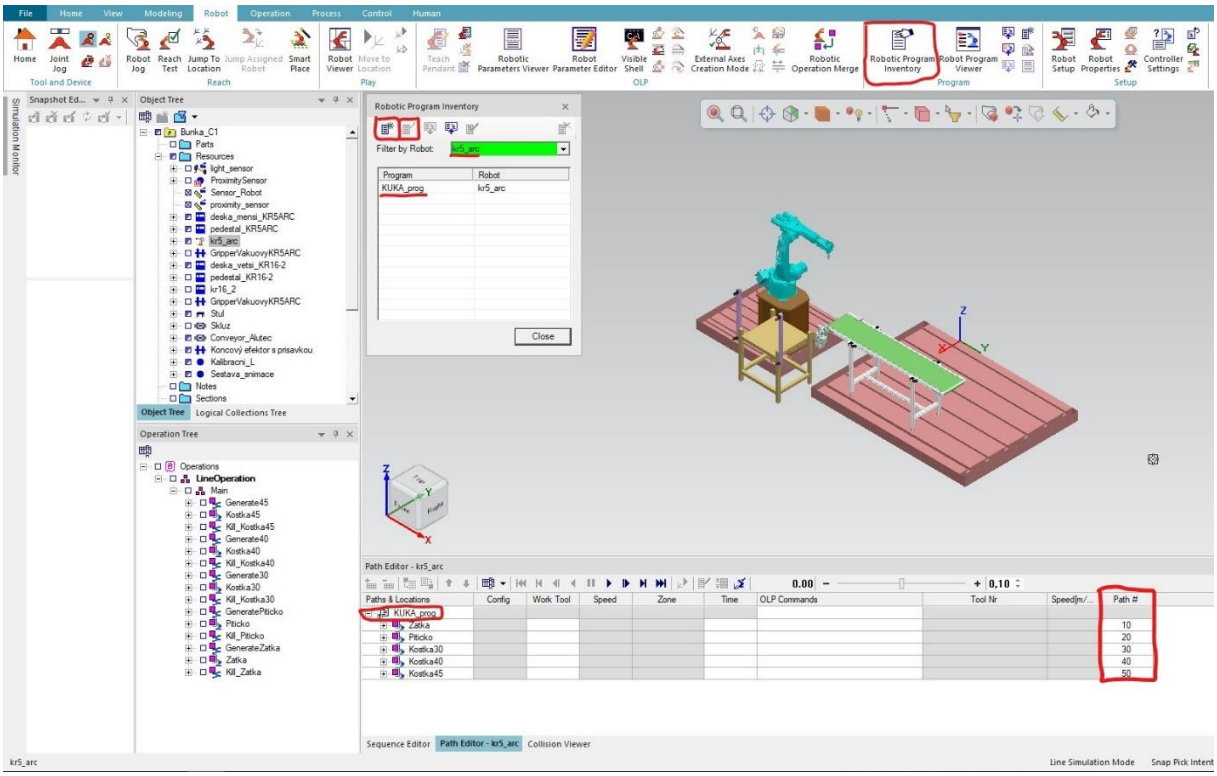
9 PROGRAM PRO ROBOT

Generování robotického programu je jednou z nejpřínosnějších funkcí Process Simulate. Programátor nepotřebuje reálný robot, na kterém by přes teach pendant vytvářel program a učil ho jednotlivé body trajektorie. Pokud má v simulaci definovaný kontroler a vytvořené dráhy pohybu, má několik možností, jak vytvořit program, který pak lze buď bez úprav nahrát do robotu a spustit nebo program upravit a vytvořit v něm logiku nebo proměnné, ovládající další příkazy.

9.1 Generování programu v Process Simulate

Program tvořený jednou cestou lze generovat kliknutím pravým tlačítkem myši na danou operaci v okně *Operation Tree* a vybráním možnosti *Download to Robot*. Program je pak uložen do vybrané složky, odkud se dá zkopírovat do reálného robotu.

Dalším, komplexnějším způsobem je otevření okna *Robotic Program Inventory* přes stejnojmenný příkaz v menu *Robot* a sekci *Program*. V okně lze ikonou Create New Program, umístěnou vlevo nahoře, vytvořit nový program. Pokud je použito více robotů, lze zvolit, pro který robot je program vytvářen. Dále lze vedlejším tlačítkem *Open in Program Editor* program otevřít v *Path Editoru,* kde je možné do programu jednoduchým přetažením z okna *Operation Tree* přidávat jednotlivé dráhy. Dále je zde možné ve sloupci *Path* přidat číslo programu, které lze využít v simulaci při řízení pohybů robotu. Po dokončení úprav lze program stáhnout opět příkazem Download to Robot, který se v okně *Robotic Program Inventory* nachází vlevo uprostřed. Druhý způsob generování programu e uveden na obr. 55.



OBR. 55 GENEROVÁNÍ PROGRAMU Z PROCESS SIMULATE

9.2 Úprava robotického programu ve WorkVisual

Vygenerovaný robotický program se sestává ze dvou souborů, což je typické pro programovací jazyk KUKA KRL. Prvním souborem je soubor .dat, obsahující definici všech použitých bodů. Druhým souborem je .src, který obsahuje samotný program. Po vygenerování syrového programu z Process Simulate však program v souboru .src obsahuje pouze uspořádané body jednotlivých trajektorií, vytvořených při simulaci a neobsahuje žádnou logiku. Je tedy nutné oba soubory upravit.

Úpravu je možné provést v jakémkoli textovém editoru nebo IDE, ideální je však použít software WorkVisual od firmy KUKA. Jednou z výhod je implementace jazyka KRL, díky čemuž je program neustále upravován a programátor je vždy upozorněn na chyby v kódu. Další výhodou je možnost připojení se ke kontroleru robotu a stažení potřebných dat z kontroleru. Lze také přiřadit jednotlivé příkazy ke konkrétním komunikačním portům, pokud má robot komunikovat například s nadřazeným PLC. Z těchto důvodů je tento software použit.

Při úpravě programu je dobré nejdříve upravit soubor.dat. Úprava spočívá v promazání nepotřebných bodů, které byly vytvořen pouze pro potřeby simulace a zpřehlednění zbylých definic bodů.

Následuje úprava souboru .src, kde je robot nejdříve poslán do Home pozice. Následuje pohyb do bodu odebrání součásti z dopravníku. Tento bod má proměnné souřadnice X, Y a podle typu výrobku také Z. Je proto nutné zavést proměnné pro tyto hodnoty přiřadit je souřadnicím definovaného bodu. Následuje přesun k mezibodu, který je použit kvůli možným kolizím. Poté je robot poslán do místa položení součásti na odkládací stůl. Pozice tohoto bodu ve všech osách souřadného systému se liší podle typu manipulované součásti. Je proto použit příkaz Switch, který je řízen proměnnou z PLC, která obsahuje informaci o typu dílu. Na závěr se robot vrátí do Home pozice. Při vytváření této logiky je potřeba přesunout potřebné body trajektorie z původních drah na správné řádky kódu a nepotřebné body promazat.

Dále je potřeba přiřadit jednotlivé proměnné ke konkrétním komunikačním portům a provést přepočet hodnot posílaných z PLC, jelikož KUKA zvládá přijímat pouze proměnné typu BOOL a různé varianty INT. V případě INT hodnot jsou navíc hodnoty v rozmezí od - 10 do 10, jelikož INT hodnoty jsou posílány přes analogovou kartu. Tyto hodnoty jsou přepočteny pomocí násobku 32 767, což je poloviční hodnota INT a vyděleny 10, což je hranice napětí analogové I/O karty.

Na závěr je program nahrán do kontroleru a lze přejít k testovací fázi. Tu lze provést na reálném pracovišti nebo program opět nahrát do Process Simulatu a provést simulaci. V takovém případě je však nutné provést potřebné úpravy simulace a komunikace s virtuálním PLC.

10 TCP/IP KOMUNIKACE PLC A KAMERY

Pro přenos informací mezi PLC a kamerou je potřeba použít real time komunikaci. Vzhledem k omezením na straně softwaru kamery je nejvhodnějším komunikačním protokolem TCP/IP. Tento druh komunikace funguje na principu serveru a klienta, přičemž server může být vytvořen externím programem a PLC i kamera by se v takovém případě k serveru připojily jako klienti. Jednodušším řešením však je server vytvořit přímo v PLC.

10.1 Nastavení TCP/IP serveru v PLC

Pro možnost použití TCP/IP komunikace a k tomu určených knihoven je nejdříve nutné nainstalovat doplněk TF6310-TCP-IP, který lze stáhnout ze stránek firmy Beckhoff. Následně je potřeba přiřadit licenci a importovat knihovnu TCP/IP obdobným způsobem jako v kapitole 7.1.1.

Pro definici serveru je dobré založit nový funkční blok s vhodným pojmenováním. Tento funkční blok pak musí být volán jinou funkcí nebo minimálně funkcí *Main*. V tomto funkčním bloku je pak volána funkce *F_CreateServerHnd*, ve které je definována IP adresa lokálního zařízení v síti a komunikační port serveru. Dále je použit funkční blok *fbServerClientConnection*, který zajišťuje propojení serveru s klientem. Je tedy nutné v něm definovat IP adresu klienta a opět komunikační port.

Pro odesílání zpráv slouží funkční blok *fbSocketSend*, ve kterém je definována proměnná *sSentData*, která je typu string a slouží k zápisu zprávy pro odeslání. Samotné odeslání zprávy řídí proměnná bSend, která je typu BOOL. Přijímání zpráv funguje obdobně, pomocí funkční ho bloku *fbSocketReciev* a proměnný *sRecievedData* a *bRecieve*. Tvorbu serveru popisuje obr. 56.

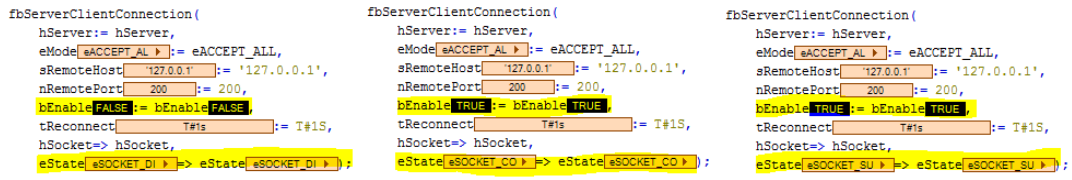
```
IF bInitialize THEN
    bInitialize := FALSE;

    F_CreateServerHnd(
        sSrvNetID:= '',
        sLocalHost:= '127.0.0.1',
        nLocalPort:= 200,
        nMode:= 0,
        bEnable:= FALSE,
        hServer:= hServer);
END_IF

fbServerClientConnection(
    hServer:= hServer,
    eMode:= eACCEPT_ALL,
    sRemoteHost:= '127.0.0.1',
    nRemotePort:= 200,
    bEnable:= bEnable,
    tReconnect:= T#1S,
    hSocket=> hSocket,
    eState=> eState);

fbSocketSend(
    sSrvNetId:= '',
    hSocket:= hSocket,
    cbLen:= SIZEOF(sSentData),
    pSrc:= ADR(sSentData),
    bExecute:= bSend);

fbSocketReciev(
    sSrvNetId:= '',
    hSocket:= hSocket,
    cbLen:= SIZEOF(sRecievedData),
    pDest:= ADR(sRecievedData),
    bExecute:= bRecieve);
```

OBR. 56 DEFINICE TCPIP SERVERU

10.2 Test komunikace

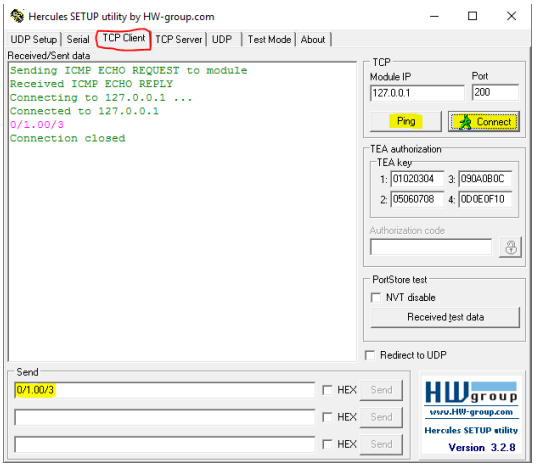
Při použití více zařízení samotným testováním serveru je dobré ověřit, zda jsou zařízení navzájem zjistitelná v síti pomocí ping testu. Správnou funkci serveru však lze zkontrolovat pouze v rámci jednoho zařízení, kde tento test není nutný. Nastavení serveru je v takovém případě jednodušší při nastavení IP adresy zařízení, jelikož lze použít univerzální adresu 127.0.0.1, tzv. local pool nebo local host. Testování je dobré provádět na IPC firmy Beckhoff.

Po nastavení adresy a portu lze spustit celý program. Po spuštění má proměnná *bEnable* v *fbServerClientConnection* hodnotu FALSE a proměnná eState ukazuje stav eSOCKET_DISCONNECTED. Je tedy nutné server aktivovat přepnutím proměnné bEnable do stavu TRUE. Proměnná eState změní svůj stav na *eSOCKET_SUSPENDED*, což značí, že server je aktivovaný a čeká na připojení klienta. Po připojení klienta se stav této proměnné změní na *eSOCKET_CONNECTED*. Celý proces popisuje obr. 57.



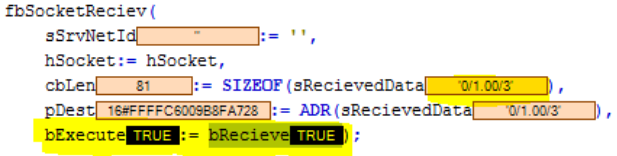
OBR. 57 SPUŠTĚNÍ SERVERU

Pro testovací účely lze vytvořit klienta pomocí externí aplikace, například programu Hercules. Jedná se jednoduchý testovací program, kde v záložce TCP Client stačí vyplnit IP adresu v poli *Module IP* a port ve stejnojmenném poli. Před připojením k serveru lze otestovat viditelnost serveru pomocí ping testu. Samotné připojení probíhá pomocí tlačítka Connect. Zprávu lze poslat v sekci Send. Ukázka nastavení klienta je na obr. 58.



OBR. 58 KLIENT V PROGRAMU HERCULES

Přijetí zprávy ze strany serveru probíhá ve funkčním bloku *fbSocketReciev* aktivací proměnné *bExecute*. Zpráva je pak uložena do proměnné *sRecievedData*. Pro přijetí další zprávy v pořadí je nutné proměnnou *bExecute* restartovat, jak ukazuje obr. 59.



OBR. 59 PŘIJETÍ ZPRÁVY

11 PROGRAM PRO PLC

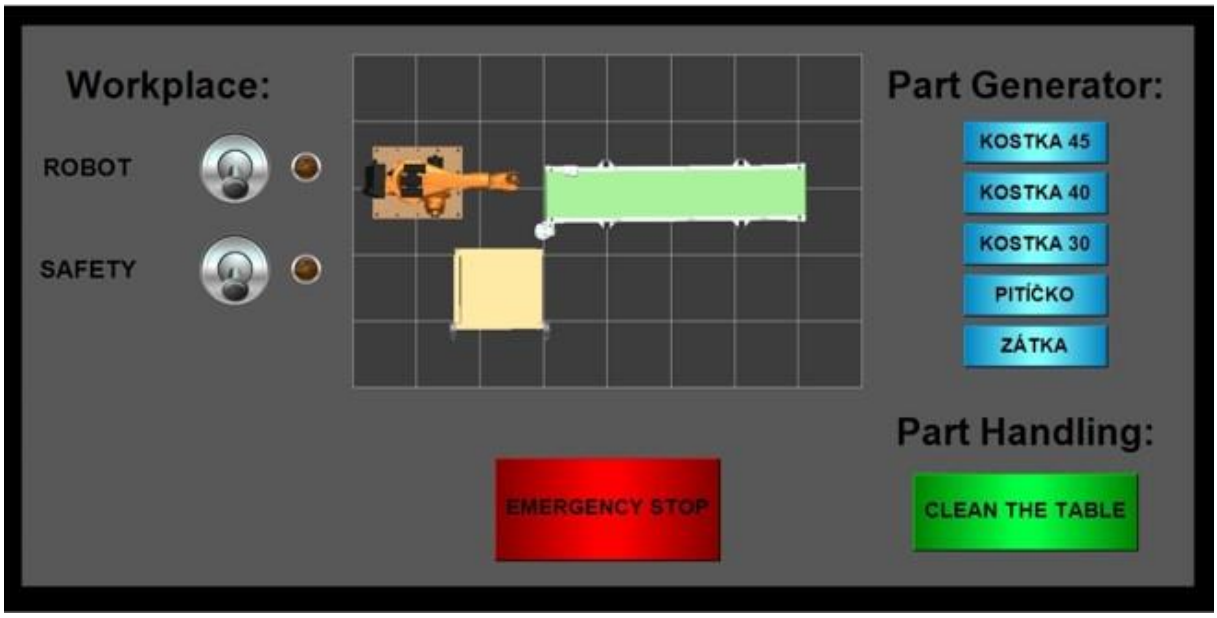
Jelikož se konečná simulace pracoviště v několika ohledech liší od reálného pracoviště, je potřeba vytvořit pro simulaci zvláštní PLC program, který by odpovídal jejím specifikacím. Program pro reálné pracoviště se pak bude lišit od simulačního značně lišit. Všechny PLC programy jsou tvořeny v prostředí TwinCAT 3 od firmy Beckhoff.

11.1 Program pro simulaci

Základem PLC programu pro simulaci je definice proměnných, uvedená v kapitole 7.1. Použité proměnné slouží k ovládání dopravníku, posílají příkazy robotu o zvoleném programu, předávají signály ze snímačů a zajišťují generování a mizení součástí pro manipulaci.

Po definici proměnných je potřeba vytvořit logiku ovládání dopravníku a řízení robotických programů. Tato logika není složitá a každý ovládaný předmět lze řídit jediným programem či funkčním blokem. Zároveň je vytvořena jednoduchá simulace bezpečnosti pomocí bezpečnostních přepínačů pro celé pracoviště, robot a emergency stopu.

Na závěr je vytvořena jednoduchá vizualizace pro ovládání pracoviště, zobrazená na obr. 60. V levé části se spouští simulace zabezpečení. V pravé části jsou umístěna tlačítka pro generování součástí a pod nimi tlačítko pro odebrání součástí ze stolu. Ve spodní části vizualizace se pak nachází emergency stop.



OBR. 60 VIZUALIZACE PLC PRO SIMULACI

11.2 Program pro pracoviště

Program nejdříve musí zpracovat hodnoty z kamery dodané ve STRING proměnné, jak je popsáno v kapitolách 10.1 a 10.2. Tato proměnná obsahuje informaci o poloze těžiště předmětu, natočení předmětu kolem osy Z, typ předmětu a informaci, zda se jedná o shodný či neshodný kus. Do těchto jednotlivých proměnných je potřeba příchozí proměnnou rozdělit. K tomu slouží příkazy pro úpravu stringu, které jsou součástí standardní knihovny.

Po vytvoření jednotlivých souřadnic je potřeba je dále připravit, aby bylo možné údaje poslat do kontroleru robotu. Ten přijímá pouze datový typ INT, jak je popsáno v kapitole 9.2. Proto je nutné souřadnice rozdělit na dva stringy, přičemž jeden obsahuje celou část čísla a druhý desetinnou část vynásobenou tisícem. Díky tomu lze obě tyto proměnné převést na INT a poslat do robotu, kde budou opět složeny, tentokrát do datového typu REAL.

Dále je třeba vytvořit logiku pracoviště, která obsahuje spouštění a zastavování dopravníku pomocí signálů z optoelektronických snímačů a zpětné vazby z robotu o dokončení programu. Signál z koncového snímače u robotu také ovládá přijímání zprávy z kamery.

Jelikož je program tvořen pro již existující pracoviště, musí brát v potaz již vytvořené proměnné a jejich přiřazení vstupům a výstupům. Také je nutné respektovat ostatní logiku pracoviště a případně ji upravit pro účely pracoviště.

Pro zobrazení aktuálních hodnot, případně zobrazení chybových stavů je vytvořena vizualizace. Skrze ni a pracovní panel lze také pracoviště ovládat.

12 PRÁCE NA REÁLNÉM PRACOVIŠTI

Aby bylo možné pracoviště uvést do provozu, je kromě tvorby programů, zmíněných v kapitolách 9.2 a 11.2, komunikací a návrhu koncového efektoru navrhnout další díly pro správnou funkci pracoviště. Jedná se především o držáky snímačů, odrazek a v neposlední řadě také kalibrační kusy pro sjednocení souřadných systémů kamery a robotu. Všechny zmíněné díly jsou navrženy v softwaru Solidworks, kde byla také vytvořena jejich výrobní dokumentace.

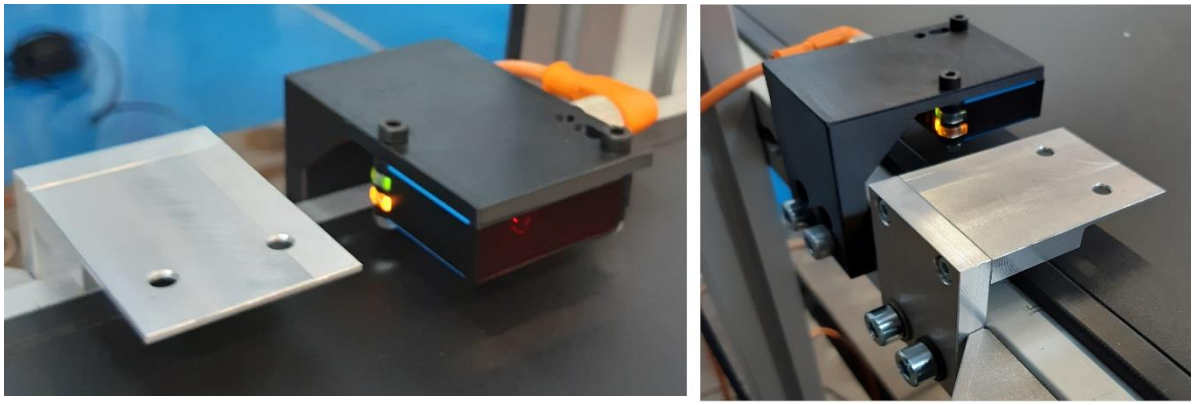
Před výrobou finálních dílů však je potřeba vytvořit prototypy. Ty jsou vyrobeny pomocí 3D tisku na tiskárně od firmy Průša, konkrétně modelu MK3S. Jako filament byl je zvolen materiál ASA černé barvy a o tloušťce struny 1,75 mm od společnosti Fillamentum. Jedná se o modifikované ABS, upravené pro zlepšení vlastností při tisku. Ve většině funkčních vlastností je ASA srovnatelný materiál s ABS, takže je vhodný pro průmyslové využití.

Po vytištění a vyzkoušení prototypů koncového efektoru a kalibračních kusů byly tyto součásti poslány do výroby. Držáky senzorů a odrazek jsou převzaty z prototypů, jelikož pro ně není potřeba použití kovových materiálů. Jejich finální verze jsou tedy vytištěny ze zmíněného plastu ASA.

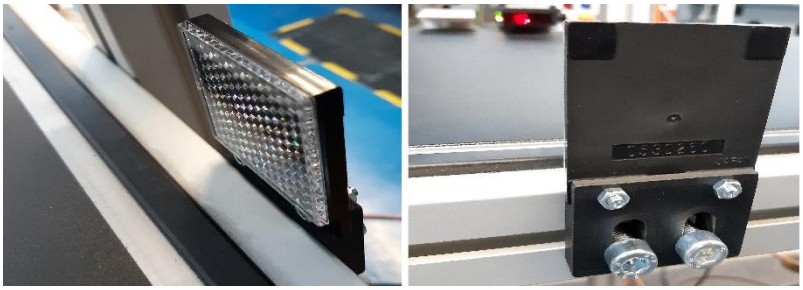
Všechny zmíněné díly jsou pak nainstalovány na pracovišti. Celé pracoviště je vyfoceno na obr. 61. Fotka nainstalovaného koncového efektoru je na obr. 62, fotka snímače a kalibračního kusu na obr. 63 a fotka odrazky na obr. 64.

OBR. 61 CELÉ PRACOVIŠTĚ

OBR. 62 NAINSTALOVANÝ KONCOVÝ EFEKTOR



OBR. 63 NAINSTALOVANÝ SNÍMAČ A KALIBRAČNÍ KUS



OBR. 64 NAINSTALOVANÁ ODRAZKA

13 ZHODNOCENÍ A DISKUZE

Návrh koncového efektoru, virtuální zprovoznění, zprovoznění komunikací a převod programů z virtuálního prostředí do reálného je komplexní problém, kterým se většinou zabývá tým lidí, specializovaných na jednotlivé kroky. Z tohoto důvodu a také z některých softwarových omezení bylo nutné některé cíle a kroky práce v průběhu tvorby změnit.

Největší takovou změnou byla samotná podstata virtuálního zprovoznění pracoviště. Reálné pracoviště přijímá informace ze softwaru, který vyhodnocuje snímky z kamery. Tyto informace jsou pak zpracovány a použity pro řízení robotu, konkrétně pro sejmutí náhodně položeného a orientovaného dílu z dopravníku. Software Process Simulate však neumožňuje přepsání souřadnic bodu odebrání dílu z dopravníku. V části virtuálního zprovoznění tedy tento úkon musel být vynechán. Virtuální zprovoznění bylo použito především pro ověření dostupnosti robotu do všech potřebných míst, generování drah robotu a následně generování programu, který bylo dále potřeba upravit. Tento problém lze vyřešit doprogramování plug-inu třetí stranou nebo přímo dodavatelem softwaru, firmou Siemens.

Dále při návrhu koncového efektoru bylo počítáno s určitým způsobem kalibrace souřadných systémů a koncový efektor tomu byl uzpůsoben. V průběhu práce však bylo zjištěno, že existuje jednodušší a do jisté míry přesnější způsob kalibrace, proto by byl nyní efektor vytvořen trochu odlišně a jednodušeji.

Dalším bodem, který byl oproti očekávání komplikovaný, je přenos souřadnic bodu odebrání dílu do kontroleru robotu. Tento problém nakonec měl řešení několikanásobným dělením a převodem proměnných, nicméně pokud by robot umožňoval lepší komunikaci, programování pracoviště by bylo mnohem jednodušší.

Největším úspěchem a přidanou hodnotou práce bylo zprovoznění a popis komunikací OPC UA a především TCP/IP v rámci PLC s následným zpracováním dat získaných z kamery.

Dalším krokem by mělo být testování reálného pracoviště a jeho zprovoznění. V této fázi se mohou vyskytnout určité problémy, které bude nutné dořešit. Většinou se jedná o komunikaci mezi jednotlivými zařízeními, kdy bývá nevhodně definovaná nebo přiřazená proměnná.

V případě požadavku návrhu pracoviště s kamerovou detekcí, a řízením robotu daty touto detekcí získanými, pomocí virtuálního zprovoznění v prostředí Process Simulate by při reálné aplikaci bylo nutné kontaktovat firmu Siemens s požadavkem na tvorbu doplňku, který by odstranil výše popsaný problém nebo tvorba tohoto doplňku vlastní cestou. Při častém použití takovéto aplikace se tento způsob řešení může finančně vyplatit. Pro jednotlivé projekty však zatím virtuální zprovoznění takového pracoviště není vhodně připraveno.

Při zhodnocení práce je také nutné vzít v potaz určení pracoviště. Jelikož se jedná o výukové pracoviště, nepočítá se s další manipulací výrobků po konečné manipulaci robotem. Z tohoto důvodu také jsou jednotlivé díly uloženy pouze na jednoduchý stůl a ne na dopravníkové paletky, které by při implementaci v rámci výrobní linky bylo určitě dobré použít. Dále by také musela být řešena bezpečnost, případně další požadavky pracoviště podle norem ČSN EN ISO 12100 a ČSN EN ISO 10218, zabývajících se robotizovanými pracovišti, případně dalšími potřebnými normami. Povaha řešené úlohy a především vybavení reálného pracoviště však tuto problematiku vyřazuje, proto také bezpečnost nebyla v této práci řešena.

14 ZÁVĚR

Cílem diplomové práce bylo virtuální zprovoznění pracoviště v dílnách ÚVSSR. K tomu bylo potřeba navrhnout nový koncový efektor a další díly. Po návrhu dílů bylo potřeba vytvořit model pracoviště pro ověření layoutu. Dále v programu Process Simulate upravit model pracoviště a vytvořit program pro robot a poté pro PLC. Všemu pak předcházela rešerše dané problematiky.

Práce je rozdělena do několika částí. V teoretické části je provedena rešerše, která uvádí problematiku práce a seznamuje se základy jednotlivých problémů, řešených v práci. Druhá část je praktická a zabývá se samotným návrhem, zprovozněním a programováním pracoviště.

Teoretická část je zaměřena na seznámení se s robotickými pracovišti a jejich variantami. Následně objasňuje problematiku robotů se sériovou kinematikou a jejich koncových efektorů. Dále popisuje PLC a způsoby jejich programování. V závěru rešerše je pojednání o virtuálním zprovoznění.

Praktická část začíná popisem požadavků na pracoviště, včetně úkonů, které na něm mají být prováděny a objektů, se kterými je manipulováno. V úvodu praktické části je také řešen layout celého pracoviště. Poté jsou popsány jednotlivé stroje použité na pracovišti a řídící jednotky těchto strojů a celého pracoviště. Dále praktická část pokračuje návrhem koncového efektoru a popisem jeho jednotlivých dílů.

Po mechanické části následuje softwarová, kde je nejdříve popsána OPC UA komunikace mezi programy Process Simulate a TwinCAT 3. Zde je popsána tvorba severu, připojení se k serveru a pro případnou potřebu také použití externího softwaru pro propojení více zařízení. Tato část byla vytvořena nad rámec cílů práce, je však s prací úzce spojena.

Následuje samotné virtuální zprovoznění. V této sekci je popsána úprava pracoviště, definování řídících prvků, tvorba drah robotu a generování robotického programu v softwaru Process Simulate. Dále je popsána úprava programu v softwaru WorkVisual od firmy KUKA.

Další kapitola popisuje TCP/IP komunikaci mezi PLC a kamerou. Kamerová část nebyla součástí práce, avšak v PLC je nutné vytvořit server, ke kterému se připojuje software vyhodnocující snímky z kamery. Kapitola řeší také testování správné funkce serveru. Tato kapitola byla vytvořena také nad rámec cílů práce.

Na závěr práce popisuje PLC programy, které bylo nutné rozdělit na dvě části. Jedna byla použita pro simulaci a druhá pro reálné pracoviště. Tento krok do jisté míry snižuje přínos virtuálního zprovoznění, nicméně bylo nutné ho podniknout z důvodů popsaných v kapitole 12.

Všechny hlavní cíle práce tedy byly splněny, přestože v průběhu došlo k několika vynuceným změnám oproti původnímu plánu. Nad rámec cílů práce byl proveden podrobný popis komunikací OPC UA a TCP/IP pro prostředí TwinCAT 3.

15 SEZNAM POUŽITÝCH ZDROJŮ

[1] Treotham Bin Picking [online]. *[cit.* 2021-03-05]. Dostupné z: http://www.treotham.com.au/media/sensors/bin-picking-pick-the-right-3d-sensor/

[2] Temaco [online]. Slušovice, *2016 [cit.* 2021-03-03]. Dostupné z: http://www.temaco.cz

[3] Artweld: Robotické svařování Artweld. *Www.artweld.cz* [online]. [cit. 2021-02-19]. Dostupné z: https://artweld.cz/roboticke-svarovani-artweld/

[4] Škoda [online]. [cit. *2021-03-03]*. Dostupné z: https://www.skoda-storyboard.com

[5] KOLÍBAL, Zdeněk. Roboty a *robotizované výrobní technologie*. První vydání. Brno: Vysoké učení technické v Brně - nakladatelství VUTIUM, 2016, 787 stran : ilustrace (převážně barevné), portréty. ISBN 9788021448285.

[6] ABB [online]. [cit. *2021-03-06]*. Dostupné z: https://new.abb.com/

[7] KUKA [online]. 2021 [*cit. 2021-03-06]*. Dostupné z: https://www.kuka.com/

[8] Fanuc [online]. 2021 [*cit. 2021-03-06]*. Dostupné z: https://www.fanuc.eu/

[9] Schunk [online]. [cit. *2021-03-05]*. Dostupné z: https://schunk.com

[10] Mattech [online]. [cit. *2021-03-05]*. Dostupné z: https://mattech.cz/

[11] Vakuový ejektor. FESTO [online]. *FESTO Group* [cit. 2021-03-14]. Dostupné z: https://www.festo.com/cms/cs_cz/9830.htm

[12] PLC-automatizace [online]. Praha: *PLC-automatizace [cit.* 2021-02-19]. Dostupné z: http://plc-automatizace.cz

[13] Odborné časopisy. Odborné časopisy [*online]. 2014* [cit. 2021-02-22]. Dostupné z: http://www.odbornecasopisy.cz/elektro/clanek/vyuziti-inteligentniho-komunikacniho-systemu-vede-ke-snizeni-nakladu-na-vyrobu-rozvadece--1891

[14] Siemens [online]. Siemens [*cit. 2021-02-19]*. Dostupné z: https://new.siemens.com

[15] Beckhoff [online]. Verl: *Beckhoff [cit.* 2021-02-19]. Dostupné z: https://www.beckhoff.com/

[16] Allen Bradley [online]. Allen *Bradley [cit*. 2021-02-19]. Dostupné z: https://ab.rockwellautomation.com

[17] Krono-safe. Www.krono-safe.com [*online]. 16,* Avenue Carnot - 91300 MASSY, France, 2017 [cit. 2021-02-19]. Dostupné z: http://www.krono-safe.com/deterministic-real-time-communication-paradigm/

[18] PLC Academy [online]. [*cit. 2021-02-19]*. Dostupné z: https://www.plcacademy.com

[19] BRAŽINA, Jakub. Virtuální zprovoznění *výrobního systému [*online]. Brno, 2019 [cit. 2021-05-08]. Dostupné z: https://www.vutbr.cz/studenti/zav-prace/detail/116785. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav výrobních strojů, systémů a robotiky. Vedoucí práce Jan Vetiška.

[20] BEDNÁŘ, Martin. Návrh experimentálního *robotického pracoviště pro* manipulační operace. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2018, 75 s. Vedoucí diplomové práce Ing. Aleš Pochylý.

[21] KUKA KR 5 arc. Arc *welding and* more [online]. Gersthofen, Germany: KUKA ROBOTER GMBH, (5), 4 [cit. 2021-03-14]. Dostupné z: https://robolution.eu/letoltes.php?type=media&name=1219-kuka-kr-5-arc-robot-adatlap.pdf

[22] KUKA KR C4. Www.kuka*.com [online*]. KUKA AG, 2021 [cit. 2021-03-20]. Dostupné z: https://www.kuka.com/en-gb/products/robotics-systems/robot-controllers/kr-c4

[23] Snímače SICK. Www.sick.com *[online]*. SICK AG, 2021 [cit. 2021-03-14]. Dostupné z: https://www.sick.com/cz/cs/w11-2/wl11-2p2430/p/p241256

[24] OPC Foundation - What is *OPC*. *OPC* Foundation [online]. 16101 N. 82nd Street, Suite 3B Scottsdale, AZ 85260-1868 USA: OPC Foundation [cit. 2021-04-09]. Dostupné z: https://opcfoundation.org/about/what-is-opc/

[25] All about TMC Files. Filext *[online]*. Gardena, California [cit. 2021-04-10]. Dostupné z: https://filext.com/file-extension/TMC

16 SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK

UVSSR	Ústav Výrobních Strojů a Robotů
PLC	Programmable Logic Controller
IPC	Industrial PC
CPU	Central Processing Unit
I/O	Input/Output
ST	Structured text
LD	Ladder Diagram
FBD	Function Block Diagram
XML	Extensible Markup Language
TCP	Tool center point
OPC	Open Platform Communications
DA	Data Access
UA	Unified Architecture
XML	Extensible Markup Language
TCP	Transmission Control Protocol
IP	Internet Protocol
TMC	Trade Manager Catalog
GVL	Global Variable List
CEE	Cyclic Event Evaluation
IDE	Integrated Development Environment

16.1 Seznam tabulek

16.2 Seznam obrázků

16.3 Seznam rovnic

17 SEZNAM PŘÍLOH